การหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว
TRACKING 3D POSITION BY INERTIAL MOTION TRACKER

จักรวาล ทองมี
JAKKAWAN THOGMEE
ณัฏฐทรี ตั้งชื่นจริญ
NATTATONG SUNGCHAROEN
พงศกร จันทึกเอก
PHONGSAKON CHANTHINOK

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว
Tracking 3D Position by Inertial Motion Tracker

นายจักรวาล ทองมี
Jakkawan Thongmee

นายณัฐพงศ์ สังข์เจริญ
Nattapong Sungcharoen

นายพงศกร จันทีนอก
Phongsakon Chanthinok

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

Tracking 3D Position by Inertial Motion Tracker

Jakkawan Thongmee

Nattapon Sungcharoen

Phongsakon Chanthinok

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

หัวข้อปริญญานิพนธ์	การหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว
รายชื่อนักศึกษา	นายจักรวาล ทองมี รหัสนักศึกษา 53010188
	นายณัฐพงศ์ สังข์เจริญ รหัสนักศึกษา 53010477
	นายพงศกร จันทีนอก รหัสนักศึกษา 53011017
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2556
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผู้ช่วยศาสตราจารย์บุณย์ชนะ ภู่ระหงษ์

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

.....

(ผู้ช่วยศาสตราจารย์บุณย์ชนะ ภู่ระหงษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว
รายชื่อนักศึกษา	นายจักรวาล ทองมี รหัสนักศึกษา 53010188
	นายณัฐพงศ์ สังข์เจริญ รหัสนักศึกษา 53010477
	นายพงศกร จันทีนอก รหัสนักศึกษา 53011017
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2556
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผู้ช่วยศาสตราจารย์บุณย์ชนะ ภู่ระหงษ์

บทคัดย่อ

โครงงานนี้แสดงถึงการสร้างอุปกรณ์เพื่อทำการวัดการเคลื่อนที่และเคลื่อนไหว ของวัตถุที่เราสนใจ และนำค่าที่วัดได้ไปจำลองการเคลื่อนที่และเคลื่อนไหวของวัตถุในโปรแกรมจำลองที่พัฒนาขึ้นเอง อุปกรณ์ที่สร้างขึ้น มีส่วนประกอบที่สำคัญคือ เซนเซอร์วัดความเร่งการเคลื่อนที่ เซนเซอร์วัดความเร็วเชิงมุม เซนเซอร์เข็มทิศ ไมโครคอนโทรลเลอร์ และโมดูลไวไฟ หลักการทำงานง่ายๆคือ เมื่อวัตถุเกิดการเคลื่อนที่หรือเคลื่อนไหว เซนเซอร์วัดความเร่งการเคลื่อนที่ เซนเซอร์วัดความเร็วเชิงมุม จะวัดค่าได้ ไมโครคอนโทรลเลอร์ส่งข้อมูลผ่านไวไฟไปยังคอมพิวเตอร์ เพื่อคำนวณโดยโปรแกรมที่พัฒนาขึ้นซึ่งมีค่าจากเซนเซอร์เข็มทิศเป็นตัวช่วยอ้างอิง จากนั้นแสดงผลออกมาในรูปแบบของการจำลองพฤติกรรมและลักษณะการเคลื่อนที่เคลื่อนไหวของวัตถุ ในส่วนของการทดลองนั้นมีการทดลองโดยติดกับรองเท้าแล้วทำการเดิน ติดกับรถบังคับวิทยุแล้วบังคับให้รถวิ่งในรูปแบบต่างๆ และจำลองว่าการเคลื่อนไหวในอริยาบทต่างๆของตัวอุปกรณ์เอง ซึ่งผลการทดลองที่ได้นั้น ในส่วนของการวัดระยะยังมีความคลาดเคลื่อนอยู่บ้างในระดับเล็กน้อย แต่สามารถวัดและแสดงพฤติกรรมการเคลื่อนที่และเคลื่อนไหวของวัตถุได้อย่างชัดเจน

Thesis Title	Tracking 3D Position by Inertial Motion Tracker		
Student	Jakkawan	Thongmee	Student ID. 53010188
	Nattapong	Sungcharoen	Student ID. 53010477
	Phongsakon	Chanthinok	Student ID. 53011017

Degree	Bachelor of Engineering
Program	Information Engineering
Year	2013
Thesis Advisor	Asst.Prof. Boonchana Purahong

ABSTRACT

This project is about creating the device to measure the movement and motion of the objects which we are interested. After that, the value will be imitated to the program which we have developed. The major components of our device are Accelerometer, Gyroscope, Digital Compass, Microcontroller and Wifi Module. The easy process is. When there is the movement of the objects, Accelerometer and Gyroscope will evaluate the value. Next, Microcontroller sends object's information through Wifi Module to the computer for calculating by our program, and the compass sensor as their reference. Then displays them in the form of a motion model behavior and movement of objects. In experiments, The experiments with shoes and then walk. Adjacent to the RC (Radio Control Car), forcing the car in another ways. And simulate the movements in the various chapters of the device itself. The experimental results of them. The measurements are also some minor discrepancies. It can measure and display the mobile habits and movements of objects clearly

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยความอนุเคราะห์ช่วยเหลือจาก อาจารย์บุณย์ชนะ ภู่ระหงษ์ ที่ช่วยให้คำชี้แนะและแนวคิดในการสร้างผลงานขึ้นมา

ขอบคุณพ่อแม่ ที่เลี้ยงดูและมอบกำลังใจให้กับพวกเรา จนทำให้พวกเราสามารถมีวันนี้ได้

ขอบคุณเพื่อนๆ ที่คอยชี้แนะ ให้แนวคิด ให้กำลังใจ และอยู่ด้วยกันมาตลอด ขอบคุณในมิตรภาพที่มอบให้

ขอบคุณฟาบิโอ ผู้ริเริ่มคิดอัลกอลิทึมหลายๆอย่างในปริญญานิพนธ์นี้ ที่เราได้นำมาใช้

ขอบคุณผู้มีพระคุณทุกๆคน ที่อาจจะไม่ได้กล่าวถึง ณ ที่แห่งนี้

สิ่งที่มีประโยชน์ สิ่งที่เกิดคุณค่า อันเกิดจากปริญญานิพนธ์นี้ คณะผู้จัดทำขอมอบให้แก่ผู้มีพระคุณทุกท่าน

นายจักรวาล ทองมี
นายณัฐพงศ์ สังข์เจริญ
นายพงศกร จันทีนอก

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญรูป VIII
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 จุดประสงค์ 2
1.3 ขอบเขตของโครงงาน 2
1.4 ผลที่คาดว่าจะได้รับ 2
1.5 อุปกรณ์ที่ต้องใช้ 2
1.5.1 ฮาร์ดแวร์ 2
1.5.2 ซอฟต์แวร์ 3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ 4
2.1 ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์ Arduino Nano 3.0 (MCU : AVR ATmega328P) 4
2.1.1 คุณสมบัติพื้นฐานของ Arduino Nano 3.0 5
2.1.2 คุณสมบัติพื้นฐานของ MCU : AVR Atmega328P 7
2.1.3 คำอธิบายขา 9
2.2 ทฤษฎีเกี่ยวกับ Logic Level Converter 11
2.3 เซนเซอร์วัดความเร็วเชิงมุม (Gyroscope) 12
2.4 เซนเซอร์วัดความเร่ง (Accelerometer) 15
2.4.1 มิเตอร์วัดอัตราเร่งแบบไซซมิกแมส (seismic mass accelerometer) 15
2.4.2 มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก (piezoelectric accelerometer) 17
2.5 เซนเซอร์เข็มทิศไฟฟ้า (Digital Compass) 18
2.5.1 การบอกทิศทางที่เกี่ยวข้องกับเข็มทิศไฟฟ้า 19
2.5.2 แนวคิดและหลักการทำงานของเข็มทิศไฟฟ้า 20
2.6 เซนเซอร์อินฟาเรด (Infared Sensor) 22
2.7 เทคโนโลยี Wireless Lan 23
2.7.1 ความเป็นมาของ Wireless Lan 23
2.7.2 Wireless LAN คืออะไร 24

สารบัญ (ต่อ)

หน้า

2.7.3 ลักษณะเด่นของ Wireless lan 25
2.7.4 มาตรฐาน Wireless LAN 27
2.7.5 เทคโนโลยีในการส่งสัญญาณ Wireless Lan 33
2.7.6 อุปกรณ์สำหรับ Wireless Lan 37
2.7.7 ข้อคำนึงในการเลือกซื้ออุปกรณ์ที่ใช้งานในระบบเครือข่ายไร้สาย 38
2.8 ทฤษฎีการเคลื่อนที่ 41
2.8.1 กฎการเคลื่อนที่ของนิวตัน 41
2.8.2 ระยะทาง ความเร็ว ความเร่ง 42
2.9 ทฤษฎีการหมุนของวัตถุ 45
2.9.1 ทฤษฎีควอเทอร์เนียน 45
2.9.2 ทฤษฎีมุมของออยเลอร์ 49
2.10 โปรแกรมที่เกี่ยวข้อง 51
2.10.1 Microsoft Visual Studio 51
2.10.2 ไลบรารี (Library) 55
2.10.3 3D Tools for the Windows Presentation Foundation (WPF) 56
2.10.4 Processing 57
บทที่ 3 การวิเคราะห์และออกแบบระบบ 58
3.1 การออกแบบฮาร์ดแวร์ 58
3.1.1 หลักการทำงานโมดูลเซนเซอร์ รุ่น GY-86 (MPU6050 HMC5883L MS5611) 58
3.1.2 หลักการทำงานของโมดูล Wireless Lan ที่ใช้งาน 59
3.1.3 หลักการทำงานของ Serial UART to USB mini B Converter V4.0 66
3.1.4 ตั้งค่าการเชื่อมโมดูล Wireless Lan กับ Access Point 67
3.1.5 หลักการทำงานของไมโครคอนโทรลเลอร์ Arduino Nano 3.0 68
3.1.6 หลักการทำงานและคุณสมบัติของ IR Sensor 69
3.1.7 การต่อโมดูลต่างๆเข้าด้วยกัน และกระบวนการทำงาน 70
3.2 การออกแบบโปรแกรม 75
3.2.1 การออกแบบโปรแกรมเพื่อจำลองกราฟแบบ สามมิติ 75
3.2.2 การเอียงวัตถุ สามมิติ 87
3.2.3 การวาดกราฟ สองมิติ 89
3.2.4 การวาดเส้นลง Canvas 2D 95
3.2.5 ภาพรวมการทำงานของโปรแกรม 99

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง 105
4.1 การทดลองวัดการเคลื่อนที่และเคลื่อนไหวด้วยการเดิน 105
4.1.1 ทดลองเดินบนพื้นราบ 105
4.1.2 ทดลองเดินขึ้นบันได 107
4.2 การทดลองวัดการเคลื่อนที่ด้วยการติดรถบังคับ 110
4.2.1 ทดลองวัดระยะทางและความสูง ด้วยการติดอุปกรณ์กับรถบังคับ 110
4.2.2 การทดลองบังคับรถบังคับวิทยุ ให้วิ่งอย่างอิสระ 113
4.2.3 การทดลองวัดมุม 144
4.3 ผลการทดลองจำลองการเคลื่อนไหวของวัตถุแบบสามมิติ 118
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง 121
5.1 บทสรุปโครงงาน 121
5.2 ปัญหาที่พบในระหว่างดำเนินงาน 121
5.3 แนวทางแก้ไขและพัฒนาต่อ 121
บรรณานุกรม 122
ภาคผนวก 123

สารบัญตาราง

ตารางที่ หน้า
2.1 ตารางอัตราการส่งข้อมูลจำเพาะ ของ IEEE 802.11b 29
2.2 ตารางอัตราการส่งข้อมูลจำเพาะของ IEEE 802.11a 31
2.3 ตารางเปรียบเทียบมาตรฐานของ Wireless Lan 32
2.4 ตารางเปรียบเทียบคุณสมบัติของคลื่นความถี่วิทยุและแสงอินฟราเรด 36
3.1 ตารางรายละเอียดพอร์ตเชื่อมต่อของโมดูลเซนเซอร์รุ่น GY-521 59
3.2 ตารางรายละเอียดพอร์ตเชื่อมต่อของโมดูล Wireless Lan รุ่น TNS-WII3 60
3.3 ตารางคุณสมบัติเฉพาะตัวทางไฟฟ้าของโมดูล Wireless Lan รุ่น RN-XV WiFly Module (RN-171) 61
3.4 ตารางRadio Characteristics 61
3.5 ตารางคำอธิบายขาของ Wireless Lan รุ่น RN-XV WiFly Module (RN-171) 62
3.6 ตารางรายละเอียดขาที่ใช้งานของ Serial UART to USB Mini B 67
3.7 ตารางคุณสมบัติพื้นฐานของ Arduino Nano 3.0 68
3.8 ตารางคุณสมบัติพื้นฐานของ IR Sensor โมดูล TCRT5000 and TCRT5000L 70
3.9 ตารางตำแหน่งข้อมูลและชนิดข้อมูล 103
4.1 ตารางผลการทดลองที่ 4.1.1 105
4.2 ตารางผลการทดลองที่ 4.1.2 107
4.3 ตารางผลการทดลองที่ 4.2.1 110
4.4 ตารางผลการทดลองที่ 4.2.3 114

สารบัญรูป

รูปที่ หน้า

1.1 ภาพรวมของระบบ 1
2.1 ไมโครคอนโทรลเลอร์ Arduino Nano 3.0 (MCU : AVR ATmega328P) 4
2.2 MCU : AVR ATmega328P 5
2.3 Block Diagram ของ MCU : AVR ATmega328P 6
2.4 วงจร Logic Level Converter 11
2.5 Gyroscope 12
2.6 หลักการทำงานของไจโรสโคป 13
2.7 Gimbaled Gyroscope 14
2.8 Acceelerometer 15
2.9 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมส 16
2.10 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก 17
2.11 แนวแม่เหล็กโลก 18
2.12 เข็มทิศ 18
2.13 แนวมุมทิศ (Azimuth) 19
2.14 ตัวอย่างเซนเซอร์ Digital Compass IC 20
2.15 กระบวนการเปลี่ยนค่าความต้านทาน 20
2.16 หลักการทำงานของเซนเซอร์อินฟาเรด 22
2.17 Tablet มีความสามารถเหมือนคอมพิวเตอร์โน๊ตบุ๊คแต่มีน้ำหนักเบา พกพาสะดวก 24
2.18 อุปกรณ์ Access Point 25
2.19 ระบบ Wireless Lan แบบ Infrastructure 27
2.20 แสดงการเปรียบเทียบอัตราเร็วกับระยะทางระหว่าง มาตรฐาน 802.11a และ802.11b 31
2.21 Access Point ที่มีการกระจายสัญญาณแบบ Multiple-Input Multiple-Output 37
2.22 Wireless Lan Card แบบ USB 38
2.23 ตัวอย่างการอินทิเกรตสัญญาณ 43
2.24 ตัวอย่างการอินทิเกรตสัญญาณ 44
2.25 การหมุนของวัตถุในระบบควอเทอร์เนียน 45
2.26 กายภาพของมุมโรล มุมพิตช์ และมุมยอ 49
2.27 กายภาพของการหมุนแบบลำดับ 2-1-3 50
2.28 ขั้นตอนการสร้างโปรเจค ของ Windows Form Application 52
2.29 ตัวอย่างหน้าจอ UI 53

สารบัญรูป(ต่อ)

รูปที่ หน้า

2.30 ตัวอย่างโค้ดของโปรแกรม WPF 53
2.31 โครงสร้างของ WPF ใน .NET Framework 54
2.32 ตัวอย่างการนำเข้าไลบรารี 55
2.33 ตัวอย่างการจำลองรูปแบบ 3D 56
2.34 ตัวอย่างโปรแกรม Processing 57
3.1 โมดูลเซนเซอร์ รุ่น GY-86 (MPU6050 HMC5883L MS5611) 59
3.2 โมดูล Wireless Lan รุ่น TNS-WII3 60
3.3 ขนาดทางกายภาพ (1 mil = 0.0254 millimeters) 61
3.4 การต่อ WiFly กับ Serial UART เพื่อตั้งค่า 63
3.5 การตั้งค่าโมดูล Wireless Lan ผ่านโปรแกรม Tera Term version 4.80 64
3.6 Router รุ่น Buffalo WZR-HP-G300NH 64
3.7 การตั้งค่าเร้าเตอร์เพื่อเชื่อมต่อกับ WiFly 65
3.8 การตั้งค่าเร้าเตอร์เพื่อเชื่อมต่อกับ WiFly 65
3.9 การตั้งค่าเร้าเตอร์เพื่อเชื่อมต่อกับ WiFly 66
3.10 ตำแหน่งขาที่ใช้งานของ Serial UART to USB Mini B Converter V4.0 67
3.11 การตั้งค่าโมดูล Wireless Lan ผ่าน PC 68
3.12 โปรแกรมตั้งค่าโมดูล Wireless Lan ผ่าน PC 68
3.13 Arduino Nano 3.0 69
3.14 IR Sensor โมดูล TCRT5000 and TCRT5000L 69
3.15 ลายวงจรของอุปกรณ์ที่พัฒนาขึ้น 70
3.16 การประกอบชิ้นส่วนอุปกรณ์ 71
3.17 การประกอบชิ้นส่วนอุปกรณ์ 71
3.18 การประกอบชิ้นส่วนอุปกรณ์ 72
3.19 ขนาดของอุปกรณ์เมื่อต่อทุกชิ้นส่วนเข้าด้วยกัน เทียบกับกระเป๋าสตางค์ 72
3.20 กระบวนการทำงานโดยรวม 73
3.21 Flow chart โปรแกรมของ Microcontroller Arduino Nano 74
3.22 มุม Azimuth และ Elevation 75
3.23 Coordinate Axis 77
3.24 2D Coordinate System และ 3D Coordinate System 78
3.25 โปรแกรมจำลอง 3D Coordinate System Without 3D Engine 81

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.26 รูปเมื่อทำการเพิ่มโค้ด 82

3.27 แบบจำลอง 3D 86

3.28 โปรแกรมใช้ในการทดลอง 86

3.29 การคำนวนมุมเพื่อใช้ในการวาดที่ถูกต้อง 87

3.30 การเอียงตามมุมที่ได้จากเซนเซอร์ โดยผ่านการคำนวนแล้ว 89

3.31 ซ้าย Cartesian Coordinate System ขวา Computer Coordinate System 89

3.32 รูปเมื่อทำการรันโปรแกรม 91

3.33 รูปเมื่อทำการเพิ่มเติม function AddChartStyle ลงในโปรแกรม 94

3.34 การแสดงผลของโปรแกรม เมื่อใช้จริง 95

3.35 การวาดกราฟ 2D (ด้านซ้าย) การวาดเส้น สามมิติลงบน Canvas แบบ สองมิติ 98

3.36 Flowchart ภาพรวมการทำงานของโปรแกรม 99

4.1 ติดอุปกรณ์กับรองเท้าและทดลองเดิน 106

4.2 สถานที่ทำการทดลองเดินบนพื้นราบ ระยะทาง 810 เซนติเมตร 106

4.3 ภาพจำลองสามมิติ แสดงพฤติกรรมและระยะทางการเดินบนพื้นราบ 107

4.4 ทำการทดลองเดินขึ้นบันได 108

4.5 ความสูงของขั้นบันได 18 เซนติเมตร 108

4.6 ระยะทางบนเดินต่อบนพื้นราบ 580 เซนติเมตร 109

4.7 ภาพจำลองสามมิติ แสดงพฤติกรรมและสเกลการเดินขึ้นบันได 109

4.8 รถบังคับที่ติดอุปกรณ์ 111

4.9 สนามทดลอง และระยะทางทดลองบนแผ่นสีขาว 240 เซนติเมตร 111

4.10 ความสูงของสนามทดลอง 15 เซนติเมตร 112

4.11 กราฟของโปรแกรม แสดงระยะทาง และความสูง ของรถบังคับขณะทำการทดลอง 112

4.12 สถานที่ทำการทดลองบังคับรถบังคับแบบอิสระ 113

4.13 กราฟของโปรแกรม แสดงผลการทดลองบังคับรถบังคับแบบอิสระ 112

4.14 ตั้งรถที่แกน 0 องศา 115

4.15 หมุนรถทำมุม -45 องศา กับแกน 0 องศา 115

4.16 หมุนรถทำมุม -22.5 องศา กับแกน 0 องศา 115

4.17 หมุนรถทำมุม 22.5 องศา กับแกน 0 องศา 116

4.18 หมุนรถทำมุม 45 องศา กับแกน 0 องศา 116

4.19 ผลการทดลองที่มุม -45 องศา กับแกน 0 องศา 116

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.20 ผลการทดลองที่มุม -22.5 องศา กับแกน 0 องศา 117
4.21 ผลการทดลองที่มุม 22.5 องศา กับแกน 0 องศา 117
4.22 ผลการทดลองที่มุม 45 องศา กับแกน 0 องศา 118
4.23 อุปกรณ์อยู่ระนาบปกติ ก่อนทำการเคลื่อนไหว 119
4.24 ทดลองโดยการหมุนอุปกรณ์ในแนวระนาบ 119
4.25 ทดลองโดยการตั้งอุปกรณ์ขึ้น 120

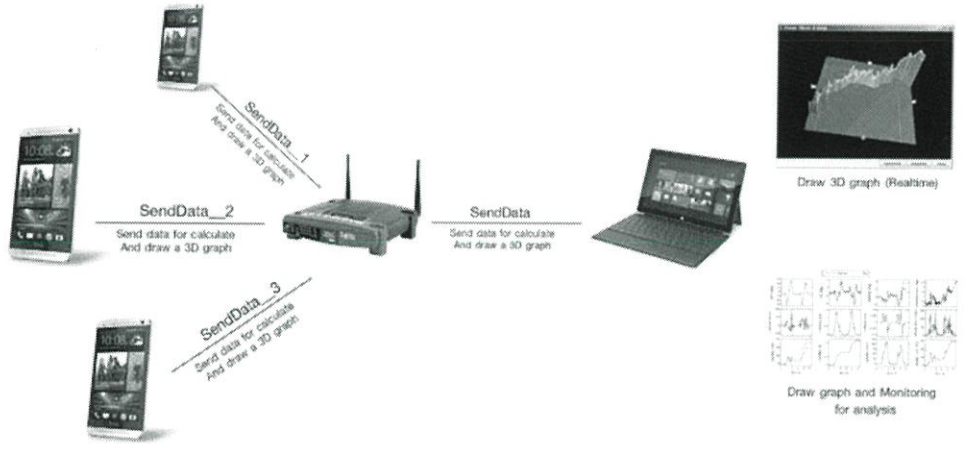
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันระบบการติดตามเป้าหมายและระบุพิกัดมีหลากหลายรูปแบบ ซึ่งแต่ละชนิดจะมีข้อดีและข้อเสียแตกต่างกันไป ตัวอย่างเช่น GPS สามารถระบุตำแหน่งตามพิกัดบนพื้นผิวโลกได้ แต่ไม่สามารถบอกตำแหน่งความสูงได้ และ ไม่สามารถใช้งานได้เมื่ออยู่ในจุดอับสัญญาณ ซึ่ง GPS เหมาะสำหรับการระบุตำแหน่งในบริเวณกว้าง หรือ การระบุตำแหน่งด้วยระบบ RFID ก็ไม่สามารถระบุการเคลื่อนที่ของวัตถุว่าเคลื่อนที่อย่างไร ทางกลุ่มจึงได้ทำโครงงานการหาตำแหน่งในสามมิติโดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว (Tracking 3D Position by Inertial Motion Tracker)

การหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว เป็นการนำความสามารถของ Accelerometers, Gyroscope และ Digital Compass มาประยุกต์เข้าด้วยกัน เมื่อนำค่าที่ได้จากการวัดขององค์ประกอบทั้งสามมาคำนวณในสมการ จะทำให้ทราบได้ว่าวัตถุมีการเคลื่อนที่อย่างไร และระบุตำแหน่งได้ จุดเด่นของระบบนี้คือ ระบุตำแหน่งได้ละเอียด ระบุได้ว่าวัตถุมีการเคลื่อนที่อย่างไร และสามารถติดตามการเคลื่อนที่ของวัตถุได้ทั้งสามแกน

การหาตำแหน่งในสามมิติโดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว สามารถนำไปประยุกต์ใช้ได้หลากหลายรูปแบบ กับการวัดและติดตามการเคลื่อนที่ของวัตถุ เช่น วัดและติดการเคลื่อนที่ของเครื่องบิน นักกีฬา UAV เป็นต้น



รูปที่ 1.1 ภาพรวมของระบบ

1.2 จุดประสงค์

ในการทำโครงงานนี้มีจุดประสงค์ในการทำงานโดยสามารถระบุได้ดังนี้

1.2.1 เพื่อศึกษาการทำงานของ Motion Sensor และ Position Sensor

1.2.2 เพื่อศึกษาและประยุกต์ใช้ Motion Sensor และ Position Sensor กับการวัดและติดตามการเคลื่อนที่ของวัตถุ

1.2.3 การหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว (Tracking 3D Position by Inertial Motion Tracker)

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถติดตามการเคลื่อนไหวของวัตถุได้

1.3.2 สามารถใช้ค่าต่างๆที่อ่านได้จากเซนเซอร์นำมาจำลองการเคลื่อนไหวในรูปแบบของสามมิติได้

1.3.3 สามารถหาตำแหน่งที่แน่นอนแม่นยำ หรือคลาดเคลื่อนอยู่ในค่าที่ยอมรับได้

1.3.4 สามารถหาอัตราเร็ว ความเร่งของวัตถุ และระยะทางที่วัตถุเคลื่อนที่ได้

1.4 ผลที่คาดว่าจะได้รับ

ผลที่คาดว่าจะได้รับ ถ้าหากโครงงานการหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว (Tracking 3D Position by Inertial Motion Tracker) สำเร็จ คือ

1.4.1 ระบบที่พัฒนาขึ้นสามารถนำไปใช้ในการระบุตำแหน่งได้

1.4.2 ระบบที่พัฒนาขึ้นสามารถนำไปใช้ติดตามการเคลื่อนที่ของวัตถุได้

1.4.3 มีความรู้ความเข้าใจในระบบการหาตำแหน่งในสามมิติ โดยใช้เซนเซอร์ตรวจจับการเคลื่อนไหว

1.4.4 มีความรู้ความเข้าใจในระบบ Motion Sensor มากขึ้น

1.4.5 สามารถนำระบบไปประยุกต์ใช้กับการติดตามวัตถุที่สนใจได้

1.5 อุปกรณ์ที่ต้องใช้

1.5.1 ฮาร์ดแวร์

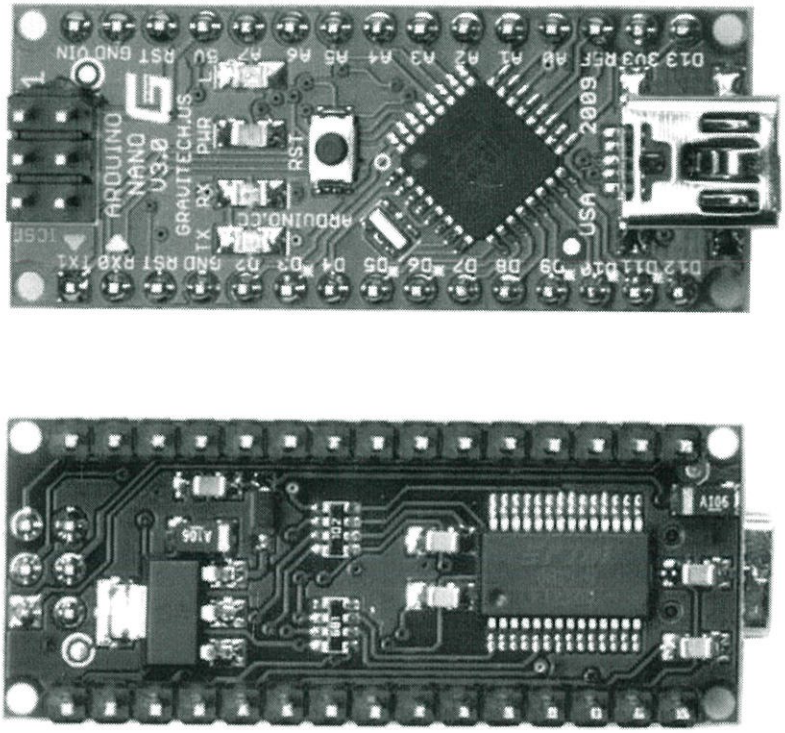
- เครื่องคอมพิวเตอร์สำหรับพัฒนา Application และโปรแกรมประมวลผลสามมิติ จำนวน 1 เครื่อง
- โมดูลเซนเซอร์ จำนวน 1 ตัว
- โมดูลไวไฟ จำนวน 1 ตัว
- ไมโครคอนโทลเลอร์ จำนวน 1 ตัว
- Air Station Wireless BUFFALO WZR-HP-G300NH จำนวน 1 เครื่อง

1.5.2 ซอฟต์แวร์

- Processing
- Visual Studio
- MATLAB
- Arduino IDE

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้

2.1 ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์ Arduino Nano 3.0 (MCU : AVR ATmega328P)

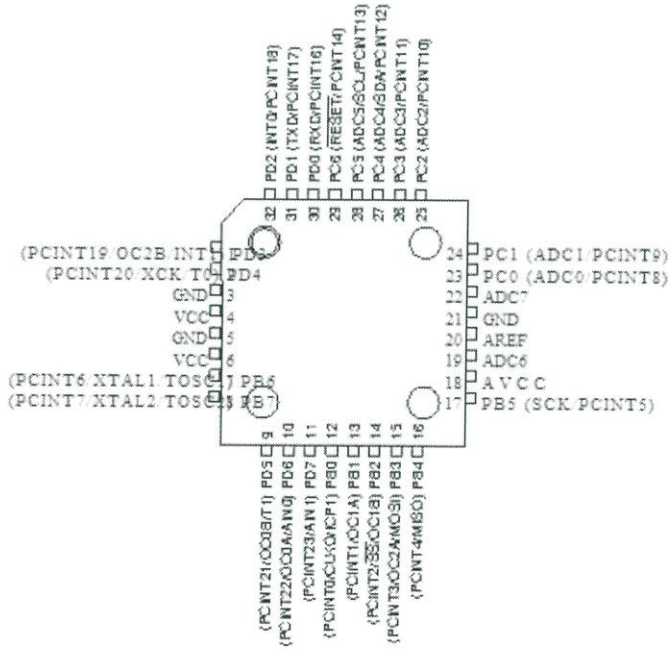


รูปที่ 2.1 ไมโครคอนโทรลเลอร์ Arduino Nano 3.0 (MCU : AVR ATmega328P) [8]

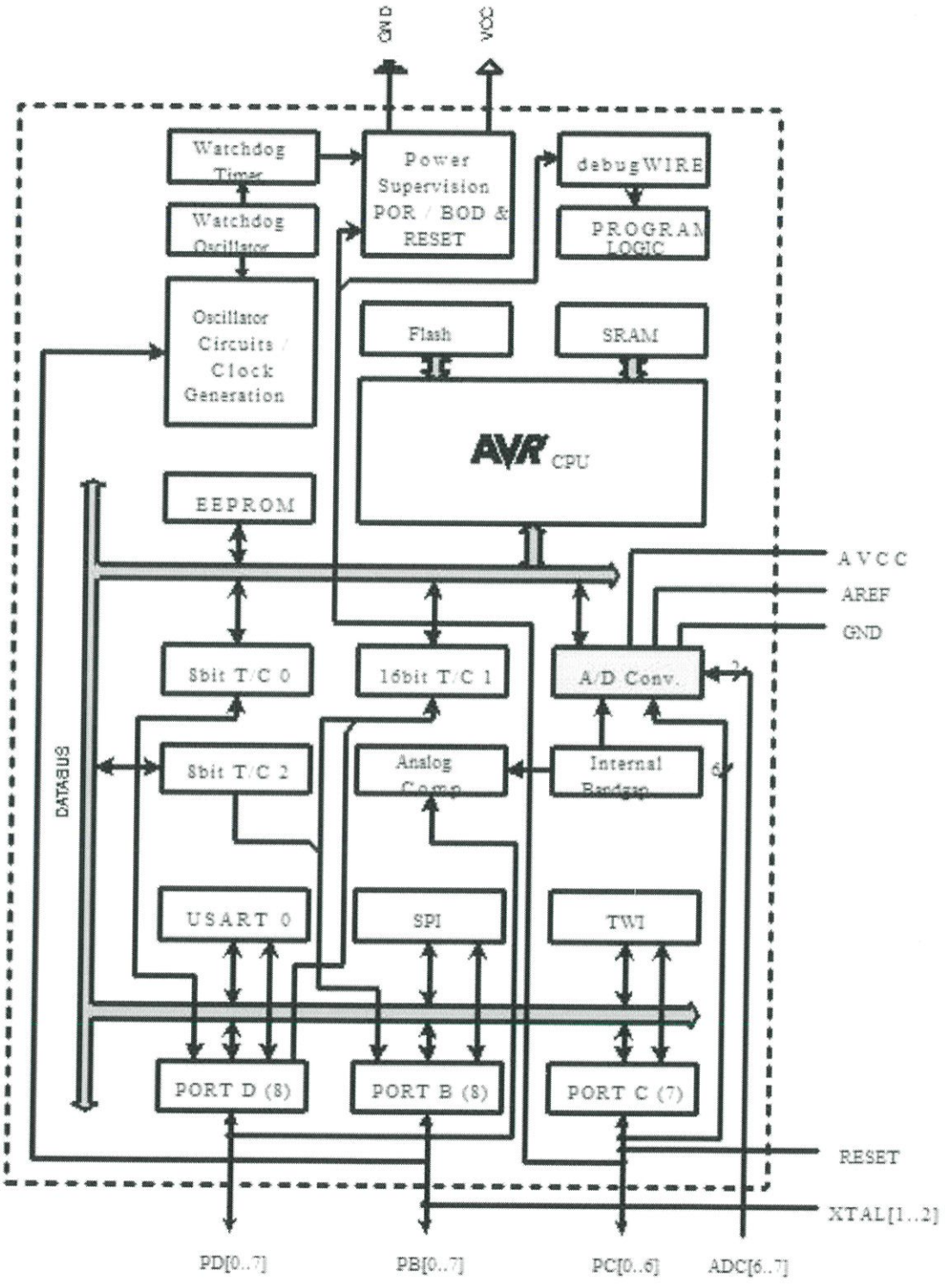
ไมโครคอนโทรลเลอร์ตัวนี้มีจุดเด่นที่ ขนาดเล็ก มีทรัพยากรพื้นฐานครบ และทำงานร่วมกับ MCU : AVR Atmega328P เนื่องด้วยราคาที่ไม่แพงและง่ายต่อการศึกษาเรียนรู้ จึงเหมาะสำหรับผู้เริ่มต้นศึกษา แต่ยังคงความสามารถที่เพียงพอต่อการทำงานทั่วไปตามความสามารถของตัวมันเองได้ และง่ายต่อการประยุกต์ใช้งานต่างๆด้วย

2.1.1 คุณสมบัติพื้นฐานของ Arduino Nano 3.0

Microcontroller	ATmega328
Operating Voltage (Logic Level)	5 V
Input Voltage (Recommended)	7-12 V
Input Voltage (Limits)	6-20 V
Digital I/O Pins	14 (of Which 6 Provide PWM Output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB (ATmega328) of Which 2 KB Used by Bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"



รูปที่ 2.2 MCU : AVR ATmega328P [7]



รูปที่ 2.3 Block Diagram ของ MCU : AVR ATmega328P [7]

2.1.2 คุณสมบัติพื้นฐานของ MCU : AVR Atmega328P

หน่วยความจำ	
หน่วยความจำแฟลช	32 กิโลไบต์
หน่วยความจำข้อมูล EEPROM	1024 ไบต์
หน่วยความจำข้อมูล SRAM	2048 ไบต์
General Purpose Registers (Accumulators)	32 ไบต์
รายละเอียด	
ความถี่สัญญาณนาฬิกา	0-20 เมกกะเฮิร์ต
Supply Voltage	1.8-5.5 V (โวลต์)
โหมด Sleep	5
Software Programmable Clock Divider	8
Hardware Multiplier	ใช่
ขา I/O	23
On Chip Oscillator	ใช่
Interrupts	26
Interrupts ขาภายนอก	26
Interrupt หรือ Wake-up บนขา Change	ใช่
Brown-Out Detection	ใช่
Power-On Reset	ใช่
Fully Static Operation	ใช่
Debug WIRE On-chip debug system	ใช่
Timers/Counters	
Timer/Counters (8 บิต)	2
Watchdog Timer ใน Chip Oscillator	1
Real Time Counter	1
Timer/Counters(16 บิต)	1
Pulse Width Modulator	6 ช่องสัญญาณ

อนาล็อก I/O	
Analog Comparator	1
Analog-to-Digital Converter (10 บิต)	6 ช่องสัญญาณ
Analog-to-Digital Converter (8 บิต)	2 ช่องสัญญาณ
พอร์ตอนุกรม I/O	
In-System Programming via พอร์ต SPI	ใช่
High Voltage Parallel Programming (12 V)	ใช่
Self-Programming via On-Chip Boot Program	ใช่
Self-Programming	
อ่านเขียนหน่วยความจำ Flash	ใช่
สัญลักษณ์ประจำตัว	
Signature Byte	Ie 92 05
Fuses	
EEPROM Preserved through chip erase	ใช่
Reset Disabled for IO Usage	ใช่
Divide External Clock by 8	ใช่
SPM Enable	ใช่
Debug WIRE Enable	ใช่
System Clock Prescaler	ใช่

2.1.3 คำอธิบายพอร์ต

1. VCC แรงดันไฟฟ้ากระแสดิจิตอล

2. GND กราวด์

3. พอร์ต B (PB7:0) XTAL/XTAL2/TOSC1/TOSC2

พอร์ต B เป็นพอร์ตอินพุท/เอาท์พุท (แบบสองทิศทาง) ขนาด 8 บิต ที่ใช้ตัวต้านทานไฟฟ้าแบบดึงกระแสจากภายในวงจร ระบบบัฟเฟอร์ของพอร์ตเอาท์พุทจะมีคุณลักษณะที่ขนานกันทั้งส่วนของ High Sink และ Source Capability ในส่วนของอินพุท ถ้าตัวต้านทานไฟฟ้าแบบดึงกระแสถูกกระตุ้น ขาของพอร์ต B ที่เป็นแบบดึงกระแสจากภายนอกจะกำเนิดกระแสไฟฟ้า แม้ว่าสัญญาณนาฬิกาจะไม่ทำงาน ขาของพอร์ต B ก็จะเป็น Tri-Stated เมื่อเงื่อนไขรีเซตเปลี่ยนเป็นแอคทีฟ

PB6 ใช้เป็นขาอินพุทไปยัง Inverting Oscillator Amplifier และ Internal Clock operating circuit ส่วน PB7 ใช้เป็นขาเอาท์พุทจาก Inverting Oscillator Amplifier โดยทั้งคู่ขึ้นกับการตั้งค่าการเลือกสัญญาณนาฬิกา หากมีการคาริเบตภายใน RC Oscillator จะใช้เป็นชิปสัญญาณนาฬิกา PB7..6 จะใช้เป็น TOSC2..1 อินพุทจากสัญญาณอะซิงโคนัส Timer/Counter2 โดยมีบิต AS2 ที่กำหนดโดย ASSR

4. พอร์ต C (PC5:0)

พอร์ต C เป็นพอร์ตอินพุท/เอาท์พุท (แบบสองทิศทาง) ที่ใช้ตัวต้านทานแบบดึงกระแสจากวงจรภายใน ระบบบัฟเฟอร์ของเอาท์พุทของขา PC..05 มีคุณลักษณะที่ขนานกันทั้งส่วนของ High Sink และ Source Capability ในส่วนของอินพุท ถ้าตัวต้านทานดึงกระแสถูกกระตุ้น ขาของพอร์ต C ที่เป็นแบบดึงกระแสจากภายนอกจะกำเนิดกระแสไฟฟ้า

5. PC6 / RESET

ถ้า RSTDISBL Fuse ถูกโปรแกรม PC6 จะถูกใช้เป็นขาของอินพุทและเอาท์พุท และคุณลักษณะอิเล็กทรอนิกส์จะแตกต่างไปจากขาอื่นของพอร์ต C

ถ้า RSTDISBL Fuse ไม่ได้ถูกโปรแกรม PC6 จะถูกใช้เป็นขาของอินพุทค่ารีเซต โดยค่า low ของขา ก็คือความยาวที่มากกว่าพัลส์ที่เล็กที่สุด แม้ว่าสัญญาณนาฬิกาจะไม่ทำงานก็ตาม

6. พอร์ต D (PD7:0)

พอร์ต D เป็นพอร์ตอินพุท/เอาท์พุท (แบบสองทิศทาง) ที่ใช้ตัวต้านทานแบบดึงกระแสจากวงจรภายใน ระบบบัฟเฟอร์ของเอาท์พุทของจา PC5..0 มีคุณลักษณะที่ขนานกันทั้งส่วนของ high

Sink และ Source Capability ในส่วนของอินพุท ถ้าตัวต้านทานดึงกระแสถูกกระตุ้น ขาของพอร์ต D ที่เป็นแบบดึงกระแสจากภายนอกจะกำเนิดกระแสไฟฟ้า

7. AV_{cc}

AV_{cc} คือขาแรงดันสำหรับ A/D Converter, PC3:0 และ ADC 7:6 จะถูกเชื่อมต่อภายนอกไปยัง V_{cc} แม้ว่า ADC จะไม่ได้ถูกใช้ก็ตาม ถ้า ADC ถูกใช้งาน มันก็จะเชื่อมต่อไปยัง Low Pass Filter

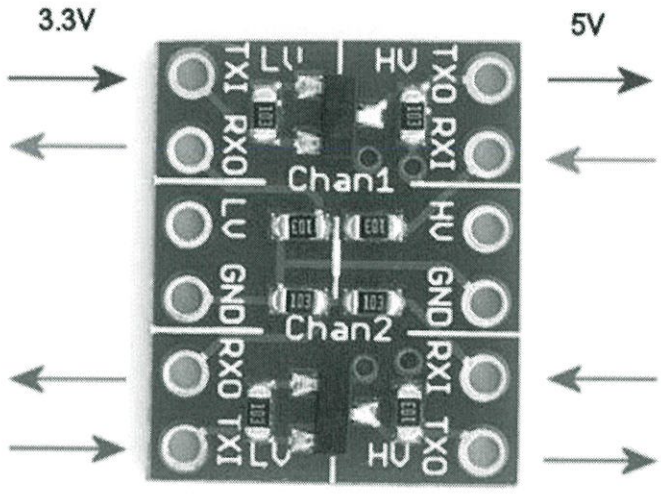
8. AREF

AREF เป็นขาอ้างอิงสัญญาณอนาล็อกสำหรับ A/D Converter

9. ADC 7:6 (TQFP และ QFN/MLF Package Only)

ในแพ็กเกตของ TQFP และ QFN/MLF ADC7:6 จะนำสัญญาณอินพุทอนาล็อก ไปยัง A/D Converter ขานี้ได้รับพลังงานมาจากช่องสัญญาณ ADC แบบ 10 บิต

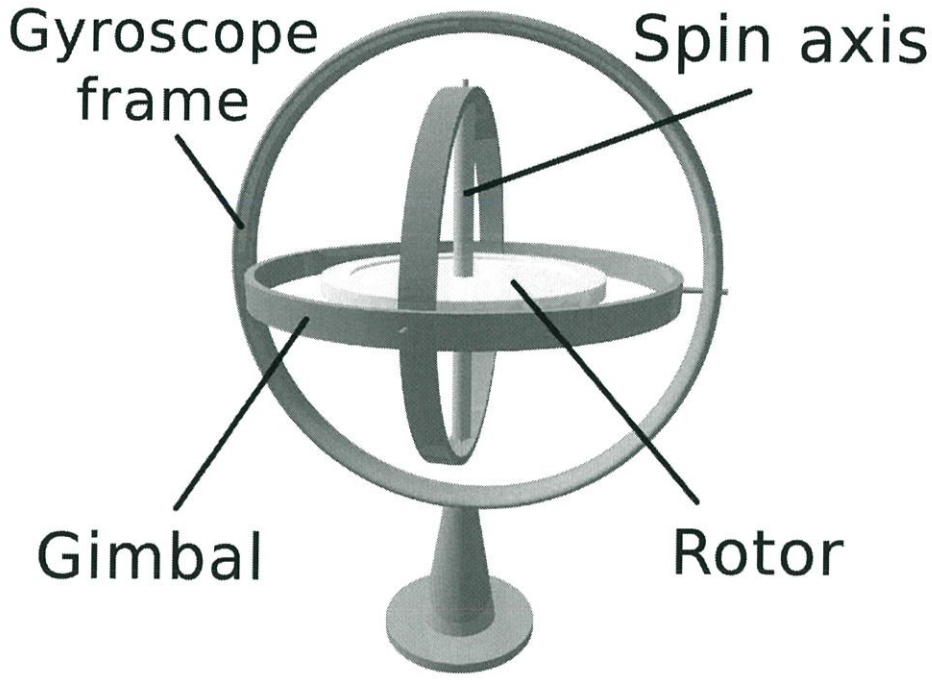
2.2 ทฤษฎีเกี่ยวกับ Logic Level Converter



รูปที่ 2.4 วงจร Logic Level Converter [16]

Logic Level Converter คือ วงจรที่ใช้ในการเชื่อมต่อสัญญาณลอจิกระหว่างอุปกรณ์ประเภท 5 V กับ อุปกรณ์ประเภท 3.3 V (หรือ 3 V) เนื่องจากอุปกรณ์แต่ละตัวใช้แรงดันอินพุทของสัญญาณไฟฟ้าไม่เท่ากัน จึงต้องทำการแปลงแรงดันสัญญาณไฟฟ้าระหว่างสองอุปกรณ์ ให้เข้ากับอุปกรณ์นั้นๆ ซึ่งอุปกรณ์บางตัว ไม่มีส่วนประกอบที่ทำการแปลงในตัวของมันเอง วิธีการทำ Logic Level Converter สามารถทำได้ทั้งการใช้วงจรสำเร็จรูป หรือ ใช้ตัวต้านทานต่อคั่นระหว่างสองอุปกรณ์ ประโยชน์ของ Logic Level Converter คือ ทำให้อุปกรณ์สองอุปกรณ์ทำงานร่วมกันได้ และเป็นตัวกำหนดทิศทางการไหลของไฟฟ้าอีกด้วย

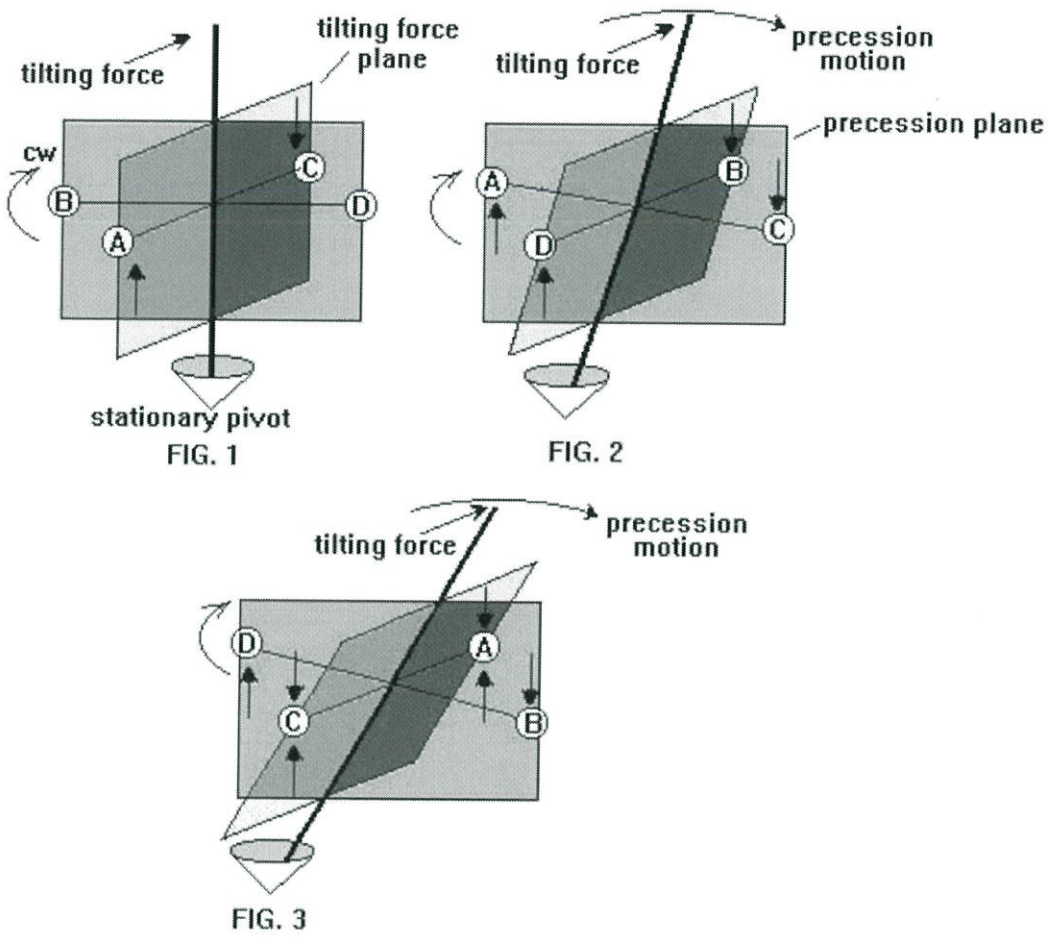
2.3 เซนเซอร์วัดความเร็วเชิงมุม (Gyroscope)



รูปที่ 2.5 Gyroscope [11]

ไจโรสโคป มีแนวคิดคือ อาศัยแรงเฉื่อยของล้อหมุน เพื่อช่วยรักษาระดับทิศทางของแกนหมุน ประกอบด้วยล้อหมุนเร็วบรรจุอยู่ในกรอบอีกทีหนึ่ง ทำให้เอียงในทิศทางต่างๆ ได้โดยอิสระ นั่นคือ หมุนในแกนใดๆ ก็ได้ โมเมนตัมเชิงมุมของล้อดังกล่าวทำให้มันคงรักษาตำแหน่งของมันไว้แม้กรอบล้อจะเอียง จากแนวคิดข้างต้นนี้สามารถนำมาประยุกต์ใช้กับ เซนเซอร์วัดความเร็วเชิงมุมได้ แนวคิดคือ เมื่อมีแรงมากระทำกับวัตถุ จนทำให้วัตถุหมุนไปรอบแนวแกน จะมีความเร็วเชิงมุมที่วัตถุนั้นหมุนไป ค่าความเร็วเชิงมุมนั้นสามารถนำมาวิเคราะห์ได้ว่า วัตถุมีการเคลื่อนที่เชิงมุมอย่างไร จากคุณสมบัติดังกล่าวทำให้สามารถนำหลักการนี้ไปประยุกต์ใช้เพื่อประโยชน์ต่างๆ มากมาย เช่น เข็มทิศ นักบินอัตโนมัติของเครื่องบิน เรือ กลไกบังคับหางเสือของตอร์ปิโด อุปกรณ์ป้องกันการกลิ้งบนเรือใหญ่ และระบบนำร่องเฉื่อย (Inertial Guidance) รวมถึงระบบในยานอวกาศ และสถานีอวกาศ

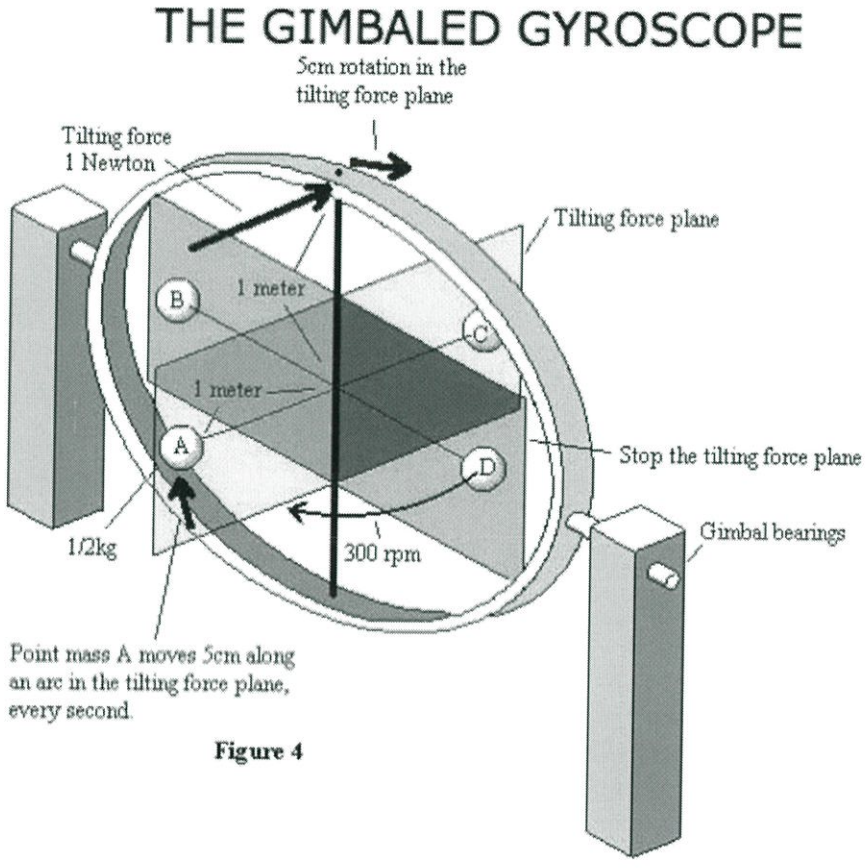
หลักการทำงานของ Gyroscope



รูปที่ 2.6 หลักการทำงานของไจโรสโคป [11]

จากรูปจะแทนที่ขอบด้วย A,B,C,D ทั้งแถบในการดูว่า Gyroscope ทำงานอย่างไร จุดล่างสุดเป็นแกนคงที่แต่สามารถหมุนได้รอบทิศทาง เมื่อแรงกระทำ (Tilting Force) ที่ส่วนบนของแกน Gyroscope จุด A จะเคลื่อนที่ขึ้นตามแนวตั้ง จุด C เลื่อนลงตามแนวนอนพร้อมกัน A และ B หมุนไป 90^0 เช่นเดียวกับที่เกิดกับ C และ D โดยที่ A นั้นยังคงเลื่อนขึ้นในตำแหน่ง 90 องศา และ C เลื่อนลง ผลของการเลื่อนของ A และ C ทำให้แกนของ Gyroscope หมุนตามการกระทำของ Precession Plane เรียกการเกิดขึ้นของลักษณะนี้ว่า Precession แกนของ Gyro จะหมุนไปทางมุมขวาเนื่องจากการหมุน ถ้า Gyroscope ถูกทำให้หมุนทวนเข็มนาฬิกามันก็จะไปทางมุมซ้าย หมายความว่าแรงกระทำตอนต้นเป็นการดึง เมื่อ Gyroscope หมุนไปอีก90 องศา ตาม Fig.3 จุด C จะมาแทนจุด A ในจุดที่แรงกระทำไปแล้วครั้งแรก การเคลื่อนที่ลงของจุด C จะถูกต้านโดย Tilting Force ทำให้แกนของ Gyroscope ไม่เปลี่ยนแปลง ยิ่งมีแรง tilting กระทำมากขึ้นแกนของ Gyroscope ก็จะมีแรงดีดกลับมามากเมื่อขอบของ Precession Plane อยู่ที่ 180 องศา จากข้างต้น

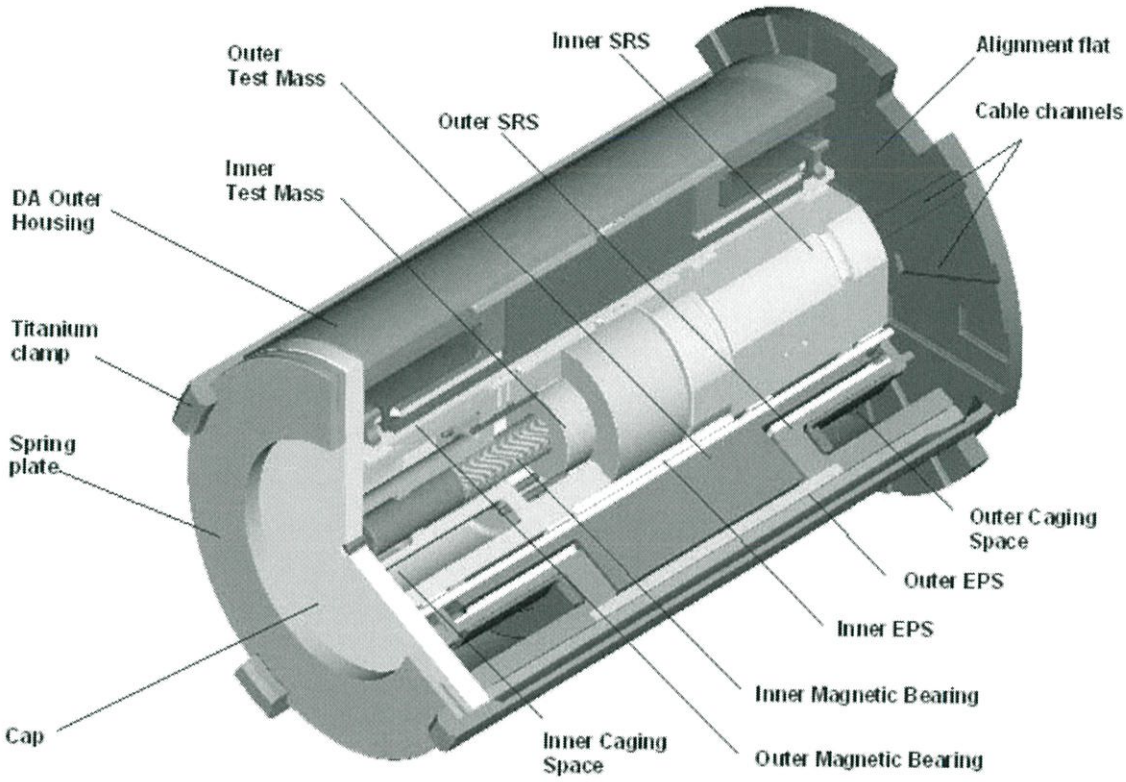
ทำให้ทราบว่าการหมุนของแกน Gyroscope เนื่องจากแรงจุด A และ C เคลื่อนที่ขึ้นลงฉะนั้นเมื่อหมุน Gyroscope ในทิศทางตรงกันข้ามกับข้างต้นจะเกิดแรงเคลื่อนที่ขึ้นลงมากขึ้น



รูปที่ 2.7 Gimbaled Gyroscope [11]

บางครั้งการเกิดขึ้นของ Precession เป็นสิ่งที่ไม่ต้องการจึงมักมีการสร้าง Gyroscope แบบมีแกนที่เรียกว่า Gimbaled Gyroscope ดังรูป ซึ่งเป็น Gyroscope พื้นฐานถูกติดตั้งไว้ในระนาบที่ตั้งฉากกับแนวแรง เมื่อหมุนขอบไปทางระนาบ Gimbal พลังงานจะถ่ายเทไปยังขอบโดย Tilting Force เพื่อหยุดการกระทำนั้น ขอบก็จะหมุนกลับในทิศทางระนาบของแรง Tilting แต่ละครั้งที่ Gyroscope ถูกกระตุ้นแกนจะหมุนไปตามส่วนโค้งในระนาบของแรง Tilting โดยที่ไม่มีการเปลี่ยนความเร็วรอบการหมุนของขอบรอบ ๆ แกน

2.4 เซนเซอร์วัดความเร่ง (Accelerometer)



รูปที่ 2.8 Acceelerometer [10]

Accelerometer เป็นมิเตอร์วัดความเร่งในหน่วย m/s^2 ซึ่งสามารถใช้เป็นตัวบ่งชี้ได้ว่าวัตถุอยู่ในสถานะนิ่งเฉย (Static) หรือ เคลื่อนที่ทันทีทันใด/หยุดทันทีทันใด (Dynamic) จากความสามารถข้างต้นนี้ Accelerometer จึงเป็นตัวบอกได้ว่าวัตถุมีการเคลื่อนที่ไปในทิศทางใด และมีค่าความเร็วเป็นเท่าไหร่

2.4.1 มิเตอร์วัดอัตราเร่งแบบไซซมิกแมส (Seismic Mass Accelerometer)

มิเตอร์ชนิดนี้อาศัยหลักการตรวจวัดระยะขจัดเชิงเส้น แล้วนำไปคำนวณหาอัตราเร่งที่เกิดขึ้น โดยเทคนิคดังกล่าวสามารถอธิบายได้ว่า วัตถุชิ้นหนึ่งจะมีความเร่งได้ จะต้องมีแรงมากระทำ ยิ่งมีแรงมากระทำมาก จะยิ่งมีความเร่งมาก ในขณะเดียวกันแรงต้านการเคลื่อนที่ก็จะมากตามไปด้วย เมื่อวัตถุเกิดการเคลื่อนที่ จะมีระยะขจัด ซึ่งก็จะแปรผันตรงกับแรงที่มากระทำที่วัตถุ ยิ่งแรงมากระยะขจัดยิ่งมาก จากความสัมพันธ์ดังกล่าวได้นำไปใช้เป็นหลักการพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมส ในการตรวจวัดอัตราเร่งของวัตถุในเทอมของระยะขจัดที่เกิดขึ้น

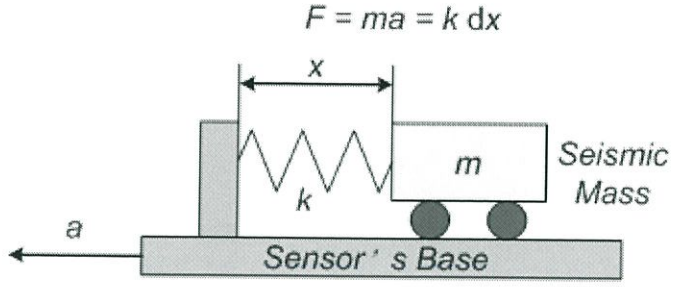


Fig-1 A spring-mass system

รูปที่ 2.9 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมส [10]

จากรูป 2.9 โครงสร้างนี้มีมวล m ที่เรียกว่ามวลตรวจการสั่นไหว (Seismic Mass) ยึดติดอยู่กับสปริงที่มีค่า Spring Constant เท่ากับ k และมวลนี้สามารถเคลื่อนที่ในแนวระดับได้ ซึ่งหลักการทำงานคือ เมื่อตัวเซนเซอร์ตัวนี้ถูกทำให้มีอัตราเร่งเกิดขึ้นจะส่งผลให้มวล m เคลื่อนที่ ซึ่งระยะที่เคลื่อนที่ออกไปจะเป็นระยะขจัดเท่ากับ x และมีทิศทางตรงกันข้ามกับการเคลื่อนที่ของตัวมิเตอร์ ดังนั้นอัตราเร่ง a ของวัตถุสามารถคำนวณหาค่าได้จากความสัมพันธ์ต่อไปนี้

$$a = xk/m$$

โดยที่ a คือ อัตราเร่งของวัตถุ หน่วย เมตร/วินาที
x คือ ระยะขจัดของมวล m หน่วย เมตร
k คือ ค่าคงที่ของสปริง หน่วย นิวตัน/เมตร
m คือ น้ำหนักของมวล m หน่วย กิโลกรัม

จากสมการดังกล่าวจะแสดงให้เห็นว่า
เมื่ออัตราเร่งของวัตถุมีค่าเพิ่มขึ้น ทำให้ระยะขจัดของมวล m มีค่าเพิ่มขึ้นตามไปด้วย
เมื่ออัตราเร่งของวัตถุมีค่าลดลง ทำให้มวล m เคลื่อนที่ไปดันสปริง
เมื่ออัตราเร่งของวัตถุหยุดลง ก็จะทำให้มวล m เคลื่อนที่กลับมาอยู่ตำแหน่งเดิม (ตำแหน่งอ้างอิง)

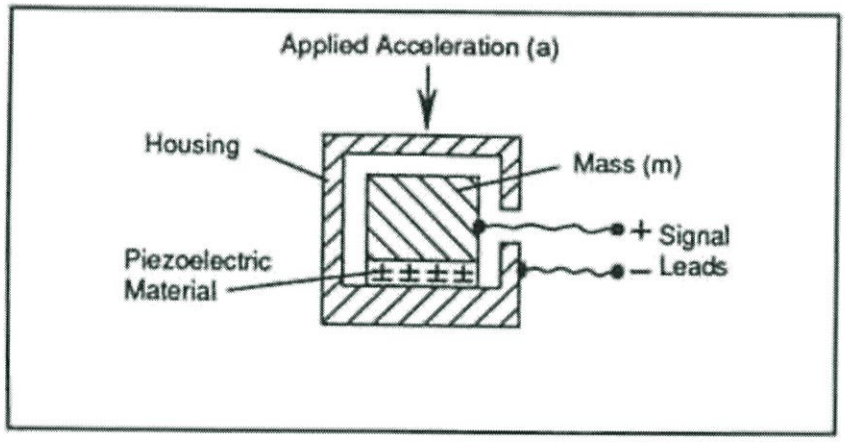
แต่ในทางปฏิบัติสามารถวัดระยะขจัดของมวล m ได้โดยอาศัยมิเตอร์อีกชนิดหนึ่ง คือ มิเตอร์วัดระยะขจัดเชิงเส้น (LVDT, Potentiometer)

ส่วนการวิเคราะห์หาค่าอัตราเร่งที่เกิดขึ้นสามารถคำนวณหาได้โดยใช้คอมพิวเตอร์ มิเตอร์วัดอัตราเร่งแบบไซซมิกแมสนี้จะนิยมใช้ในการตรวจวัดลักษณะการช็อกและลักษณะการสั่นสะเทือนที่มีความถี่ต่ำมากๆ เช่น ในเครื่องมือตรวจวัดแผ่นดินไหว หรือในเครื่องมือตรวจวัดการปะทุใต้ดินของภูเขาไฟ ฯลฯ

2.4.2 มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก (Piezoelectric Accelerometer)

คุณสมบัติพื้นฐานทางไฟฟ้าของผลึกเพียโซอิเล็กทริก (Piezoelectric Crystal) ถูกค้นพบโดย Pierre และ Jacques Curie ในราวปี ค.ศ.1880 ซึ่ง Piezoelectric Crystal นี้มันมีคุณสมบัติพิเศษคือ เมื่อถูกแรงทางกลมากระทำ จะสร้างประจุไฟฟ้าขึ้นมา โดยเป็นสัดส่วนกับแรงกระทำนั้น ซึ่งจากคุณสมบัติพิเศษนี้ได้ถูกดัดแปลงนำไปใช้สร้างอุปกรณ์ต่างๆมากมาย เช่น ใช้เป็นแบตเตอรี่จ่ายพลังงานไฟฟ้าให้กับนาฬิกาข้อมือดิจิตอลที่ใช้ทั่วไป และยังใช้สร้างมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกอีกด้วย

โครงสร้างของมอเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกจะประกอบด้วย Seismic Mass ยึดติดกับ Piezoelectric Crystal และบรรจุอยู่ในตัวถังป้องกัน โดย Piezoelectric Crystal ที่นิยมนำมาใช้งาน ได้แก่ ผลึกควอตซ์ และผลึกโซเดียมโปตัสเซียมตาเตรต(Sodium Potassium Tartrate) เพราะมีความทนทานต่อแรงกระทำ และราคาไม่แพงมากนัก



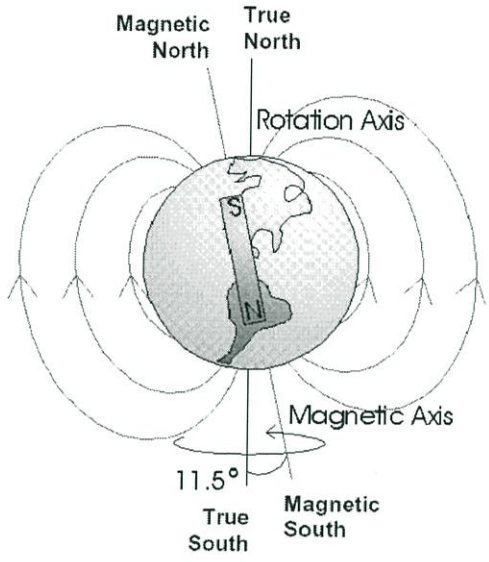
รูปที่ 2.10 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก [9]

จากรูป 2.10 สามารถอธิบายการทำงาน ได้ดังนี้

เมื่อ Seismic Mass (m) ถูกทำให้เกิดอัตราเร่ง (ถูกกด) จะส่งผ่านแรงกดไปกระทำกับ Piezoelectric Crystal ที่ถูกยึดติดอยู่ด้วยกัน ด้วยคุณสมบัติพิเศษของมันจะทำให้ประจุไฟฟ้าถูกสร้างขึ้น และถูกสายนำสัญญาณออกไปยังเอาท์พุทของวงจร โดยที่ด้านเอาท์พุทจะต้องมีวงจรขยายประจุไฟฟ้า (Charge Amplifier) เพื่อขยายค่าประจุไฟฟ้าที่ได้ให้เป็นแรงดันเอาต์พุตตามสัดส่วนของอัตราเร่งที่เกิด จะได้สามารถแสดงผลได้ด้วยโวลต์มิเตอร์

มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกตอบสนองต่อทางด้านความถี่สูงได้ดี แต่ในทางกลับกันก็จะมีผลตอบสนองทางด้านความถี่ต่ำที่ไม่ดีนัก มีขนาดค่อนข้างเล็ก น้ำหนักเบา และสามารถใช้งานที่มีอัตราเร่งได้สูงถึง 250,000 m/s^2

2.5 เซนเซอร์เข็มทิศไฟฟ้า (Digital Compass)



รูปที่ 2.11 แนวแม่เหล็กโลก [15]

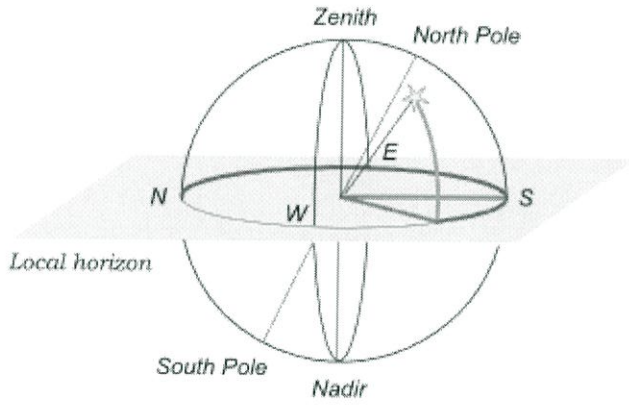
เข็มทิศ คือเครื่องมือสำหรับใช้หาทิศทาง มีเข็มแม่เหล็กที่แกว่งไกวได้อิสระในแนวนอนทอดตัวในแนวเหนือ-ใต้ ตามแรงดึงดูดของแม่เหล็กโลก และที่หน้าปัดมีส่วนแบ่งสำหรับหาทิศทางโดยรอบ เข็มทิศจึงมีปลายชี้ไปทางทิศเหนือเสมอ (อักษร N) เมื่อทราบทิศเหนือแล้วก็ย่อมหาทิศอื่นได้โดยหันหน้าไปทางทิศเหนือ ด้านขวามือเป็นทิศตะวันออก ด้านซ้ายมือเป็นทิศตะวันตก ด้านหลังเป็นทิศใต้ การบอกทิศทางในแผนที่โดยทั่วไป คือการบอกเป็นทิศที่สำคัญ 4 ทิศ คือทิศเหนือ ทิศใต้ ทิศตะวันออก และทิศตะวันตก

รูปที่ 2.12 เข็มทิศ [15]

2.5.1 การบอกทิศทางที่เกี่ยวข้องกับเข็มทิศไฟฟ้า

1. แบบมุมทิศ (Azimuth)

มุมอะซิมุท (Azimuth) คือมุมที่ใช้บอกตำแหน่งของพิกัดบนท้องฟ้า เช่น บอกตำแหน่งของดาวเพื่อตั้งกล้องดูดาว หรือบอกตำแหน่งของดาวเทียมวงโคจรค้างฟ้าเพื่อตั้งจานดาวเทียม การกำหนดมุมอะซิมุทนั้น มุมอะซิมุทจะมีค่าเริ่มที่ 0 องศาที่ทิศเหนือ และมีค่าเพิ่มขึ้นตามเข็มนาฬิกาไปมีค่า 90 องศาที่ทิศตะวันออก 180 องศาที่ทิศใต้ และ 270 องศาที่ทิศตะวันตก มุมนี้จะต้องใช้ร่วมกับ มุมอัลติจูด (Altitude) โดยมุมอัลติจูดคือมุมเงย จะมีค่าเริ่มต้น 0 องศา ที่ขอบฟ้า เงยไปจนถึง 90 องศาเมื่อเงยหน้ามองไปตรงเหนือศีรษะพอดี การใช้มุมทั้งสองนี้เรียกว่าการระบุพิกัดแบบ Horizontal Coordinate

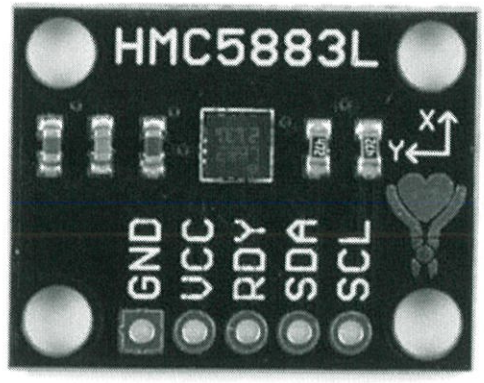


รูปที่ 2.13 แนวมุมทิศ (Azimuth) [15]

2. แบบแบริง (Bearing)

การบอกทิศทางแบบแบริง (Bearing) คือการบอกทิศทางเป็นค่าของมุมในแนวราบ ซึ่งวัดจากแนวทิศเหนือหลักไปยังแนวเป้าหมายในทิศทางตะวันออกหรือตะวันตก หรือวัดจากแนวทิศใต้หลักไปแนวเป้าหมายทิศตะวันออกหรือตะวันตก ดังนั้นขนาดของมุมแบริงจะมีค่าไม่เกิน 90 องศา การอ่านค่ามุมแบบแบริงจะเริ่มต้นด้วยทิศหลัก เช่นทิศทาง AB เบนจากทิศเหนือไปทิศตะวันตกเป็นมุม 75 องศา เรียกทิศทาง AB นั้นว่า มีมุมแบริง 75 องศาตะวันตก

2.5.2 แนวคิดและหลักการทำงานของเข็มทิศไฟฟ้า

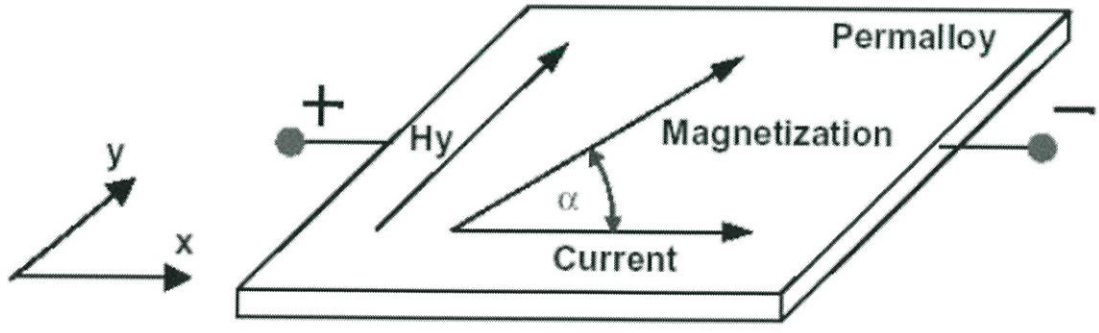


รูปที่ 2.14 ตัวอย่างเซนเซอร์ Digital Compass IC [15]

หลักการทำงานของ เข็มทิศไฟฟ้า (Digital Compass) นั้น ใช้คุณสมบัติทางแม่เหล็กของโลหะ (Magnetization) และ ปรากฏการณ์การเปลี่ยนแปลงค่าความต้านทานทางไฟฟ้าของวัตถุภายใต้สนามแม่เหล็กโลก (Magnetoresistive) โดยแนวคิดที่นำมาใช้คือ

1. Magnetization มีทิศทางคงที่โดยขึ้นอยู่กับชนิดของโลหะ
2. Magnetoresistive การเปลี่ยนแปลงค่าความต้านทานทางไฟฟ้าของโลหะ

จากคุณสมบัติทั้งสองข้อนี้ เมื่อนำมาประกอบกัน ทำให้เกิดทฤษฎี การเปลี่ยนค่าของตัวต้านทานทางไฟฟ้าของโลหะ โดย ค่าของความต้านทานจะเปลี่ยนตามมุมของ Magnetization กับทิศทางของกระแสไฟฟ้าที่ไหลผ่าน ดังรูปที่ 2.15



รูปที่ 2.15 กระบวนการเปลี่ยนค่าความต้านทาน [15]

ค่าความต้านทานที่เปลี่ยนไปสามารถคำนวณได้จากสมการ

$$R = R_0 + \Delta R \cos^2\alpha$$

$$\alpha = 0^\circ \Rightarrow R_{max}$$

$$\alpha = 90^\circ \Rightarrow R_{min}$$

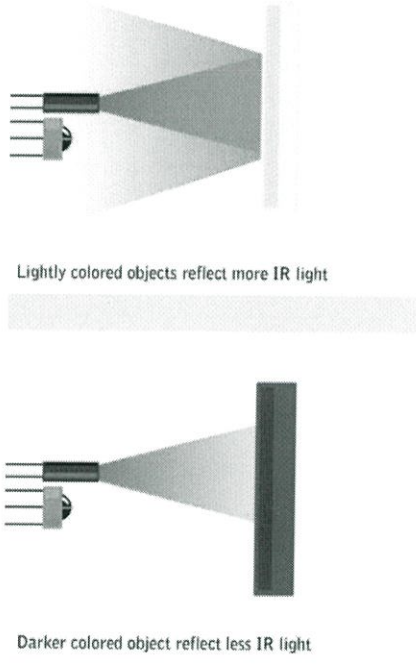
โดย α คือมุมระหว่าง Magnetization กับ ทิศทางของกระแส

จากแนวคิดหลักการทำงานของเข็มทิศ และ คุณสมบัติทางแม่เหล็กทั้งสองข้อ เมื่อนำมารวมกัน สามารถสร้างเป็น Digital Compass IC ได้ โดยการวัดความดันไฟฟ้าทางขาออก โดยการหาค่าอ้างอิงระหว่างมุมต่างๆของ α เทียบกับค่าความต้านทาน และ แรงดันทางขาออก จะทำให้หาค่าของมุมตามหลักเข็มทิศได้

2.6 เซนเซอร์อินฟาเรด (Infared Sensor)

อินฟาเรด คือ คลื่นแม่เหล็กไฟฟ้าที่มีความถี่อยู่ในช่วง 1011 – 1014 เฮิรตซ์ หรือความยาวคลื่น 10^{-3} – 10^{-6} เมตร เรียกว่า รังสีอินฟราเรด หรือเรียกอีกอย่างหนึ่งว่า คลื่นความถี่สั้น (Millimeter Waves) ซึ่งจะมีย่านความถี่คาบเกี่ยวกับย่านความถี่ของคลื่นไมโครเวฟอยู่บ้าง วัตถุร้อน จะแผ่รังสีอินฟราเรดที่มีความยาวคลื่นสั้นกว่า 10^{-4} เมตรออกมา อินฟาเรดเดินทางเป็นเส้นตรง ไม่สามารถผ่านวัตถุทึบแสง และสามารถสะท้อนแสงในวัสดุผิวเรียบได้เหมือนกับแสงทั่วไปใช้มากในการสื่อสารระยะใกล้

หลักการทำงานของเซนเซอร์อินฟาเรด



รูปที่ 2.16 หลักการทำงานของเซนเซอร์อินฟาเรด [17]

อินฟาเรดจะสะท้อนกลับมากในพื้นผิวสีขาว และสะท้อนกลับน้อยในพื้นผิวสีทึบ จากหลักการนี้จึงใช้อินฟาเรดร่วมกับเซนเซอร์รับแสง เพื่ออ่านค่าความเข้มแสงที่สะท้อนกลับมา จากนั้นนำค่าที่ได้ไปประยุกต์ใช้ต่อไป ซึ่งใช้ได้หลากหลายมาก

2.7 เทคโนโลยี Wireless Lan

เมื่อกล่าวถึงเทคโนโลยีไร้สายแล้ว Wireless LAN เป็นเทคโนโลยีที่มีผู้ให้ความสนใจมาก เนื่องจากสามารถนำเทคโนโลยี Wireless LAN (WLAN) มาประยุกต์ใช้กับองค์กร สถานศึกษา รวมถึง หน่วยงานราชการ ทั้งนี้เพื่อประหยัดค่าใช้จ่ายในส่วนของการติดตั้งระบบเครือข่าย อีกทั้ง Wireless LAN มีความคล่องตัวสูงในการเคลื่อนย้ายอุปกรณ์ที่เชื่อมต่อกับระบบเครือข่าย

ในอดีต WLAN ไม่เป็นที่นิยมเนื่องจากอุปกรณ์ Wireless มีราคาสูงและมีความเร็วในการติดต่อสื่อสารประมาณ 2 Mbps (สองล้านบิตในหนึ่งวินาที) เมื่อเปรียบเทียบกับความเร็วของอุปกรณ์เครือข่ายแบบใช้สายซึ่งมีความเร็วประมาณ 100 Mbps หรือ 50 เท่า ทำให้ประสิทธิภาพของการติดต่อสื่อสารที่ใช้ WLAN ไม่ดีเท่าที่ควรเมื่อเปรียบเทียบกับระบบเดิมซึ่งใช้สาย

ปัจจุบันอุปกรณ์เครือข่าย Wireless มีประสิทธิภาพมากขึ้นโดยมีความเร็วประมาณ 600 Mbps ซึ่งเพียงพอต่อการใช้งานเป็นอย่างมาก ดังนั้นทำให้ WLAN มีความนิยมมากขึ้นเรื่อย ๆ และเป็นหนทางสำหรับบริษัทที่ต้องการระบบเครือข่ายที่มีประสิทธิภาพ รองรับการทำงานของอุปกรณ์ที่มีการเคลื่อนที่อยู่ตลอด เช่น คอมพิวเตอร์โน้ตบุ๊ค หรือ โทรศัพท์มือถือ (Smartphone)

2.7.1 ความเป็นมาของ Wireless Lan

ความต้องการใช้ระบบแลนไร้สายมีลักษณะเช่นเดียวกับระบบเซลลูลาร์โฟน หรือโทรศัพท์เคลื่อนที่ แต่ระบบแลนไร้สายต้องการเซลล์ขนาดเล็ก และเป็นเซลล์เฉพาะกิจ เป็นเซลล์ส่วนตัวที่เชื่อมกับเครือข่ายได้ ดังนั้นจึงมีความพยายามที่พัฒนาเครือข่ายแลนแบบไร้สาย เพื่อให้รองรับความต้องการของผู้ใช้ กลุ่มผู้ใช้ที่มีความต้องการใช้แลนแบบไร้สายได้แก่ ร้านค้าปลีก ที่เก็บสินค้า โรงพยาบาล ธุรกิจขนส่ง มหาวิทยาลัย ตลอดจน องค์กรภาคธุรกิจต่าง ๆ ในปี ค.ศ. 1997 สถาบัน IEEE ได้กำหนดมาตรฐานแลนไร้สายแบบเดียวกับอีเทอร์เน็ต และเป็นชุดเดียวกับ 802 โดยให้ชื่อว่า IEEE 802.11 มาตรฐานที่เกิดขึ้นในปีนั้นยังมี ข้อจำกัดในทางเทคโนโลยี จึงกำหนดระบบการรับส่งสัญญาณด้วยขนาดความเร็ว 2 Mbps ระบบแลนไร้สาย IEEE 802.11 จึงเป็นที่รู้จักกันตั้งแต่นั้นมา

ในปี ค.ศ. 1999 IEEE ได้พัฒนามาตรฐานใหม่ของแลนระบบไร้สายและให้ชื่อ มาตรฐานที่ IEEE 802.11b โดยมีการพัฒนาให้ใช้ความเร็วในการรับส่งได้ถึง 11 Mbps และเป็นแบบฟูลดูเพล็กซ์คือ รับและส่งแยกกันด้วยความเร็ว 11 Mbps จากมาตรฐาน 802.11b ที่ประกาศออกไป ทำให้มีผู้ผลิตแลน ไร้สายออกมามาก โดยเฉพาะบริษัทผู้ผลิตอุปกรณ์เครือข่ายขนาดใหญ่ทุกบริษัทให้ความสนใจและเร่งการพัฒนาปรับปรุงกันต่อไป การพัฒนาแลนไร้สาย มิได้หยุดอยู่ เพียงแค่การทำให้เชื่อมต่อถึงกันได้เท่านั้น ระบบการดูแลรักษาความปลอดภัยของ สัญญาณข้อมูลที่แพร่กระจายในอากาศ มีการวางมาตรฐานทางด้านเอ็นคริปชั่น และการสร้างระบบดูแลรักษาความปลอดภัยการเข้าถึง มีการพัฒนาระบบการ เคลื่อนย้ายเข้าสู่เครือข่ายหนึ่งไปอีกเครือข่ายหนึ่ง หรือที่เรียกว่า โรม

มิ่ง (Roaming) มีการแบ่งโหลดระหว่างเซล โดยการตรวจสอบความแรงของสัญญาณเพื่อให้ขนาดของพื้นที่ทับซ้อนกันได้

2.7.2 Wireless LAN คืออะไร

Mobile เทคโนโลยีมีการพัฒนาอย่างต่อเนื่อง โทรศัพท์เคลื่อนที่เริ่มเป็นอุปกรณ์ที่จำเป็นในชีวิตประจำวันเพราะช่วยอำนวยความสะดวกในการติดต่อสื่อสาร เช่น ในกรณีที่รถเกิดอุบัติเหตุ ผู้ขับสามารถใช้โทรศัพท์มือถือโทรขอความช่วยเหลือได้ทันท่วงที นอกจากโทรศัพท์มือถือแล้วอุปกรณ์ Personal Information Management (PIM) เช่น เครื่อง Personal Digital Assistant (PDA) เริ่มมีผู้สนใจและใช้งานมากขึ้น เช่นใช้ในการรับส่ง E-mail การจัดตารางนัดหมาย หรือ การเก็บข้อมูลการเงินส่วนตัว ปัจจุบันเครื่องคอมพิวเตอร์โน้ตบุ๊ค มีขนาดที่เล็กลง แต่มีประสิทธิภาพที่สูงขึ้น เช่น เครื่อง Tablet PC ได้มีการติดตั้งอุปกรณ์การติดต่อสื่อสารชนิด Wireless กับระบบเครือข่ายเป็นส่วนหนึ่งของเครื่อง โดยผู้ใช้งานไม่จำเป็นต้องซื้ออุปกรณ์ติดต่อสื่อสารเพิ่มเติม

รูปที่ 2.17 Tablet มีความสามารถเหมือนคอมพิวเตอร์โน้ตบุ๊คแต่มีน้ำหนักเบา พกพาสะดวก [1]

เมื่อกล่าวถึงคำว่า Mobile แล้ว นิยมใช้กับคำว่า Wireless แต่ทั้งสองคำมีความแตกต่างกันคือ

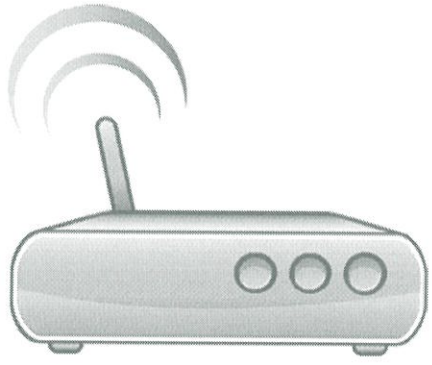
- Mobile หมายถึง ความสามารถในการเคลื่อนย้าย
- Mobile Device หมายถึง อุปกรณ์ขนาดเล็กที่สามารถพกพาได้ เช่น โทรศัพท์มือถือ เครื่อง PDA และ คอมพิวเตอร์โน๊ตบุ๊ค
- Wireless เป็นเทคโนโลยีที่ใช้ในการส่งภาพ เสียง และข้อมูล จากอีกที่หนึ่งไปอีกที่หนึ่ง โดยไม่ใช้สายไฟ แต่ใช้คลื่นวิทยุหรืออาจใช้คลื่น Infrared เป็นตัวกลางในการสื่อสาร
- LAN มาจากคำว่า Local Area Network คือระบบเครือข่ายในระยะทางใกล้ๆ เช่น ภายในห้องทำงาน การเชื่อมต่อระหว่างห้องทำงาน ระหว่างชั้น หรือ ระหว่างตึก

ดังนั้น Wireless LAN หมายถึง การติดต่อสื่อสารในระยะทางใกล้ๆ โดยไม่ใช้สายไฟและอุปกรณ์สื่อสารคือ Mobile Device หรือ คอมพิวเตอร์โน๊ตบุ๊ค

สำหรับ Wireless ที่ใช้สัญญาณวิทยุ อุปกรณ์ที่เป็นเครื่องรับข้อมูลสามารถเคลื่อนย้ายไปมาได้อย่างอิสระภายใต้รัศมีความแรงของการใช้คลื่นวิทยุ คือ การรับส่งข้อมูลมีความผิดพลาดสูง เพราะคลื่นวิทยุถูกรบกวนได้ง่ายจากสัญญาณภายนอก เช่น การเปิดเครื่องเตาอบ Microwave อาจทำให้สัญญาณ Microwave เข้าไปรบกวนระบบ Wireless ได้ ทำให้การรับส่งข้อมูลผิดพลาด ส่วน Wireless ชนิดที่ใช้สัญญาณ Infrared ในการใช้งาน อุปกรณ์เครื่องรับและเครื่องส่งต้องติดตั้งไว้ในจุดที่ไม่มีสิ่งกีดขวาง เพราะ Infrared เป็นคลื่นแสง ไม่สามารถทะลุผ่านสิ่งของได้ การทำงานของ Wireless ชนิด Infrared คล้ายการใช้ Remote Control ในการเปลี่ยนช่องของโทรทัศน์ ข้อดีของการใช้คลื่น Infrared คือ ป้องกันการรบกวนจากสัญญาณภายนอกได้ดี

Wireless ชนิดที่ใช้สัญญาณวิทยุเหมาะสำหรับใช้ในการติดต่อสื่อสารระหว่าง Mobile Device กับระบบเครือข่ายหลัก เช่น การใช้เครื่อง PDA ท่อง Web Site ภายในที่ทำงาน ผู้ใช้งานสามารถเดินไปห้องต่าง ๆ ในที่ทำงานพร้อมทั้งเรียกดูข้อมูลจากอินเตอร์เน็ตได้ตลอดเวลา สำหรับ Wireless ชนิดที่ใช้ Infrared เหมาะสำหรับการติดต่อสื่อสารระหว่างเครือข่ายหลักที่ไม่สามารถเดินสายไฟได้ เช่น การเชื่อมต่อเครือข่ายระหว่างตึกซึ่งอยู่คนละฝั่งถนน

อุปกรณ์ Access Point ทำหน้าที่รับและส่งข้อมูลกับ Mobile Devices แบบไร้สาย Mobile Devices สามารถเคลื่อนที่ภายใน Wireless Network ได้ ดังรูป



รูปที่ 2.18 อุปกรณ์ Access Point [3]

ส่วนอุปกรณ์ในระบบ Wired Network หรือที่ใช้สายไม่สามารถเคลื่อนย้ายได้ และใช้เชื่อมต่อกับคอมพิวเตอร์บนโต๊ะทำงาน แต่เมื่อใดที่ผู้ใช้งานไม่ได้อยู่ที่โต๊ะทำงาน และต้องการทำงานโดยไม่ขาดช่วง สามารถเปลี่ยนมาใช้ระบบ Wireless Network ได้โดยใช้เครื่องคอมพิวเตอร์โน๊ตบุ๊ค โดยทั่วไประบบที่ใช้ Wireless Network จำเป็นต้องมีการติดตั้งระบบ Wired Network ก่อนเพราะต้องใช้สำหรับการติดต่อสื่อสารกับระบบอินเตอร์เน็ต

2.7.3 ลักษณะเด่นของ Wireless Lan

เนื่องด้วยในปัจจุบันนี้เป็นยุคของการแลกเปลี่ยนข้อมูลข่าวสาร,แบ่งปัน (Share) ข้อมูล ,ทรัพยากรร่วมกัน โดยเฉพาะองค์กรด้านธุรกิจต่างๆ เมื่อใช้ Wireless lan ผู้ใช้สามารถ Access

ข้อมูล ได้โดยไม่ต้องมองหา ปลั๊กต่อสายแลน และผู้มีหน้า ที่ดูแล Network สามารถติดตั้ง หรือขยายการใช้งานโดยปราศจากการติดตั้งสาย หรือเคลื่อนย้ายไปตามจุดต่างๆ Wireless LAN ทำให้เกิดความสะดวก แก่ผู้ใช้ ลดค่าใช้จ่ายเรื่องสายได้ โดยเฉพาะระบบการเชื่อมโยงแบบต้องใช้สาย ซึ่งมีแลนแบบอีเทอร์เน็ตเป็นฐานใหญ่ โดยเฉพาะการเชื่อมต่อด้วยสาย UTP ผ่าน HUB ทำให้เข้าสู่เครือข่ายด้วยความเร็วสูงถึง 100 Mbps อีกด้านหนึ่งคือการเชื่อมด้วยสายโทรศัพท์โดยเฉพาะการใช้โมเด็ม และ ADSL ที่เปิดบริการกันมากอยู่ในขณะนี้

จุดเด่นของระบบแลนไร้สาย มีหลายประการ โดยเฉพาะในอดีตปัญหาทางเทคโนโลยีเป็นข้อจำกัด เพราะไม่สามารถสร้างระบบ VLSI (วงจรรวมขนาดใหญ่มาก) ที่ใช้งานย่านความถี่สูงมาก กินกำลังไฟฟ้าต่ำ มีขนาดเล็กและเบา ปัจจุบันสามารถพัฒนาวงจร CMOS ซึ่งเป็นหัวใจของการผลิตชิพที่มีวงจรซับซ้อน ให้กินกำลังงานไฟฟ้าต่ำมาก และใช้กับความถี่สูงย่าน Microwave ได้ เมื่อเป็นเช่นนี้ระบบแลนไร้สายจึงตอบสนองความต้องการเด่น ๆ ต่อไปนี้ได้

- ความสะดวกในการเคลื่อนย้าย Wireless Lan ช่วยให้ผู้ใช้สามารถ Access ข้อมูลได้ทุกๆ ที่ ในองค์กร การเคลื่อนย้ายของผู้ใช้อาจไม่เฉพาะเจาะจงอยู่ในที่ทำงานอย่างเดียว อาจครอบคลุมเลยไปยังที่ต่าง ๆ เช่น การจัดประชุมสัมมนา การเดินทาง ไปยังสถานที่ต่าง ๆ โดยผู้ใช้ติดตั้งไปเฉพาะโน๊ตบุ๊คก็สามารถเข้าถึงเครือข่ายได้ - การติดตั้ง เร็ว และ ง่าย ไม่ต้องยุ่งยากกับการลากสายเคเบิล ผ่านกำแพง ฯลฯ

- ความยืดหยุ่นของการติดตั้ง เทคโนโลยีเครือข่ายแบบแลนไร้สาย ทำให้เครือข่ายองค์กรปรับขนาดและความเหมาะสม ได้ง่าย ไม่ยุ่งยากในเรื่องการเดินสายสื่อสาร ซึ่งมีปัญหาในเรื่องสถานที่ การปรับปรุงสถานที่เพื่อเดินสายสัญญาณเป็นเรื่องไม่พึงปรารถนา เครือข่าย ไร้สายสามารถครอบคลุมพื้นที่เป็นเซลเล็ก ๆ โดยมีการเชื่อมโยงระหว่างอาคาร ได้ด้วยระบบแบบจุดไปจุด ทำให้ดำเนินการได้เร็วและสะดวกต่อการติดตั้ง

- ลดต้นทุนขององค์กร ในระยะแรกของการลงทุนติดตั้ง Wireless มี Cost ค่อนข้างสูงกว่า Wired LAN แต่ค่าใช้จ่ายโดยรวมแล้ว (รวมค่าติดตั้ง และค่าเช่าสาย) Wireless จะต่ำกว่า อย่างเห็นได้ชัด ยิ่งถ้าคิดในระยะยาวแล้วต่ำกว่ามาก

- Scalability Wireless Lan สามารถที่จะถูก Config ได้หลาย Topology เพื่อให้บรรลุตาม ความต้องการ การ Config นั้นง่ายมากตั้งแต่ แบบ Peer to Peer สำหรับผู้ใช้จำนวนไม่มาก ไปจนถึง เต็มที่ของ Infrastructure ที่มีผู้ใช้จำนวนมาก นอกจากนี้ จุดเด่นของระบบแลนไร้สายยังอยู่ที่การมีมาตรฐานที่ยอมรับกันทั่ว หมายถึงสามารถใช้งานร่วมกันได้ และเป็นมาตรฐานกลางที่กำหนดโดย IEEE มีการวางรูปแบบให้รับส่งกันได้อย่างดี โดยเฉพาะเรื่องการรักษาความปลอดภัยของคลื่นสัญญาณที่อาจถูกดักฟังได้ กรณีนี้ก็มีวิธีการเข้ารหัส การสร้างระบบเข้ารหัส ข้อมูล การให้บริการการใช้งาน และการดูแลรักษาเครือข่ายทำได้ง่ายกว่าแบบใช้สายมาก ทั้งนี้เพราะระบบได้รับการออกแบบมาให้เป็นแบบอัตโนมัติ และตรวจสอบกันเอง ระบบแลนไร้สายจึงมีจุดเด่นที่ชัดเจน และจะมีบทบาทที่สำคัญของเครือข่ายในอนาคต อันใกล้นี้

รูปที่ 2.19 ระบบ Wireless Lan แบบ Infrastructure [2]

2.7.4 มาตรฐาน Wireless LAN

มาตรฐานหลักของระบบเครือข่ายไร้สายและอุปกรณ์เครือข่ายไร้สาย คือ มาตรฐาน IEEE 802.11 เป็นมาตรฐานระบบเครือข่ายไร้สายที่ถูกกำหนดขึ้นโดย Institute of Electrical and Electronic Engineers ซึ่งเป็นองค์กรกำหนดมาตรฐานเกี่ยวกับการสื่อสารของอุตสาหกรรมคอมพิวเตอร์ โดยในส่วนมาตรฐาน IEEE 802.XX นั้นจะเป็นเรื่องเกี่ยวกับการสื่อสารผ่านเครือข่าย เช่น IEEE 802.3 ก็คือมาตรฐานของเครือข่ายแบบ Ethernet โดยในส่วนย่อย IEEE 802.11 ก็จะเป็นการสื่อสารกับเครือข่าย แต่เป็นแบบไร้สายนั่นเองมาตรฐาน IEEE 802.11 นั้นเริ่มประกาศใช้ตั้งแต่ปี ค.ศ. 1997 มาตรฐานที่เกิดขึ้นนี้ยังมีข้อจำกัดในด้าน เทคโนโลยี ซึ่งกำหนดระบบการส่งสัญญาณด้วยความเร็ว 2 Mbps และได้มีการพัฒนาเรื่อยมา โดยมีส่วนย่อยอยู่ด้วยกันถึง 9 ส่วน คือ a, b, c, d, e, f, g, h และ I โดยแต่ละชนิดนั้น ก็จะมีลักษณะหรือมาตรฐานของรายละเอียดที่แตกต่างกันไป ซึ่งหลังจาก 9 กลุ่มย่อยนี้ พัฒนามาตรฐาน IEEE 802.11 ในด้านต่างๆ จนเสร็จสิ้นแล้ว จึงได้มีการนำเอามาตรฐานที่พัฒนาเสร็จแล้วมานำเสนอและผลิตออกเป็นผลิตภัณฑ์ออกวางจำหน่าย โดยผลิตภัณฑ์ แรกที่ออกวางจำหน่ายเป็นผลิตภัณฑ์ที่พัฒนาโดยกลุ่มย่อย b จึงทำให้เกิดมาตรฐาน IEEE 802.11b ในปี ค.ศ.1999 ย่านความถี่ที่เริ่มใช้ เบื้องต้น คือ 2.4 GHz โดยมีความเร็วในการรับ-ส่งข้อมูลสูงสุดอยู่ที่ 11 Mbps ได้วางตลาดก่อนผลิตภัณฑ์กลุ่มอื่น จึงเป็นกลุ่มที่มาตรฐานได้รับการยอมรับและเป็นที่รู้จัก มากที่สุดในช่วงนี้ จากนั้นจึงตามด้วยกลุ่ม a ที่ออกความถี่สูงสุดถึง 5 GHz และมีความเร็วสูงสุดถึง 54 Mbps ใน ทั้งนี้ไม่เกี่ยวว่า a จะเก่ากว่า b และ c จะออกมาใหม่ในอนาคตตามตัวอักษร แต่จะขึ้นอยู่กับว่ามาตรฐานของกลุ่มใดทำเสร็จก่อนก็จะออกเปิดตัวก่อน โดยไม่เรียงลำดับตามตัวอักษร

- มาตรฐาน IEEE802.11

มาตรฐาน IEEE 802.11 เหมือนกับมาตรฐาน IEEE 802.3 Ethernet ซึ่งใช้กับเครือข่าย LAN แบบใช้สาย และ IEEE 802.5 สำหรับเครือข่าย Token Ring ตรงที่ มาตรฐาน IEEE 802.11

จะมุ่งความสนใจไปที่ระดับล่างสุดสองระดับของ ISO Model (คือ Physical Layer และ Data Link Layer) ซึ่งจะทำให้ Application, Network OS, Protocol, รวมทั้ง TCP/IP ใดๆก็ตามสามารถใช้งานบน 802.11 compliant WLANs ได้ง่ายๆเช่น เดียวกับใช้งานบน Ethernet โดยทั่วๆไป มาตรฐาน 802.11 นี้ใช้การส่งสัญญาณแบบคลื่นวิทยุที่ความถี่ 2.4 GHz ซึ่งเป็นความถี่ ISM (Industrial, Scientific and Medical) band สามารถส่งข้อมูลได้ด้วยอัตราความเร็ว ค่อนข้างต่ำ คือ 1 และ 2 Mbps เท่านั้นโดยใช้เทคนิคการส่งสัญญาณหลักอยู่ 2 รูปแบบ คือ DSSS (Direct Sequent Spread Spectrum) และ FHSS (Frequency Hopping Spread Spectrum) ซึ่งถูกคิดค้นมาจากหน่วยงานทหาร การส่งสัญญาณทั้ง 2 รูปแบบจะใช้ความกว้างของช่องสัญญาณ (Bandwidth) ที่มากกว่า การส่งสัญญาณแบบ Narrow Band แต่ทำให้สัญญาณมีความแรงมากกว่า ซึ่งง่ายต่อการตรวจจับมากกว่า แบบ Narrow Band หน่วยงานทหารใช้วิธีการเหล่านี้ในการปิดกั้นการใช้งานจากอุปกรณ์อื่นๆที่จะมาทำให้ระบบเกิดปัญหา โดยการส่งสัญญาณแบบ FHSS สัญญาณจะกระโดดจากความถี่หนึ่งไปยังอีกความถี่หนึ่งในอัตราที่ได้กำหนดไว้แล้ว ซึ่งจะรู้กันเฉพาะตัวรับกับตัวส่งเท่านั้น ส่วนการส่งสัญญาณแบบ DSSS จะมีการส่ง Chipping Code ไปกับสัญญาณแต่ละครั้งด้วย ซึ่งจะมีเฉพาะตัวรับกับตัวส่งเท่านั้นที่จะรู้ลำดับของ Chip สำหรับการใช้งานระบบเครือข่ายแบบไร้สายทุกวันนี้ DSSS มีคุณสมบัติที่โดดเด่นและให้ Throughput ที่มากกว่า เมื่อเร็วๆนี้เองที่ได้มีการพัฒนาจนได้อัตราการส่งข้อมูล 11 Mbps ผ่านการส่งแบบ DSSS และเป็นมาตรฐานที่โดดเด่นของ WLAN ผลิตภัณฑ์ซึ่งรองรับมาตรฐาน 802.11b (อัตราส่งถ่ายข้อมูลสูง 11 Mbps) นี้สามารถทำงานร่วมกับผลิตภัณฑ์ซึ่งทำงานกับมาตรฐาน DSSS แบบเก่า 802.11 (อัตราส่งถ่ายข้อมูล 1 และ 2 Mbps) ได้ แต่ ระบบ FHHS จะถูกใช้กับอุปกรณ์ที่มีกำลังส่งต่ำ, เป็น Application ที่ใช้งานในย่านต่ำๆ เช่น โทรศัพท์ไร้สายความถี่ 2.4 GHz แต่จะใช้งานร่วมกับผลิตภัณฑ์ DSSS ไม่ได้

• มาตรฐาน IEEE802.11b

มาตรฐาน IEEE 802.11b ซึ่งเป็นมาตรฐานระบบเครือข่ายไร้สายที่ได้รับการยอมรับมากที่สุดในโลกเพราะมีการเปิดตัวก่อนมาตรฐานอื่นและมีผลิตภัณฑ์ออกวางจำหน่ายแล้วมากและแพร่หลายที่สุดมาตรฐาน IEEE 802.11b นั้นล่าสุดได้รับการตั้งชื่อใหม่ว่า WiFi โดยได้รับการรับรองมาตรฐานและกำหนดรายละเอียดโดยกลุ่ม WECA หรือ Wireless Ethernet Compatibility Alliance ที่ประกอบด้วยสมาชิกจากผู้ผลิตในอุตสาหกรรมคอมพิวเตอร์ชื่อดังอย่าง 3com, Cisco Systems, Intersil, Agere Systems, Nokia และ Symbol Technologies ซึ่งปัจจุบันก็ยังมีสมาชิกจากบริษัทต่างๆ อีกกว่า 110 บริษัทเข้าร่วมอยู่ในมาตรฐานนี้ สำหรับรายละเอียดด้านคุณสมบัติ ของ IEEE 802.11b จะสามารถรับ-ส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps โดยใช้ความถี่คลื่นวิทยุที่ 2.4 GHz ใช้เทคนิคการส่งสัญญาณแบบ DSSS โดยย่านความถี่ที่ใช้เป็น ISM (Industrial, Scientific and Medical) Band จากระดับความเร็วที่ค่อนข้างต่ำ คือทำได้เพียง 11 Mbps เท่านั้น เมื่อเทียบกับ ระบบ LAN แบบมีสาย ที่มาตรฐานปัจจุบัน อยู่ที่ระดับ 100 Mbps

และล่าสุดมาตรฐานความเร็ว 1 Gbps กำลังเป็นที่ยอมรับและนิยมใช้งานมากขึ้นเรื่อย ๆ ก็จะเห็นว่า IEEE 802.11b นั้นค่อนข้างช้ากว่ามาก ไม่เพียงเท่านั้น คลื่นความถี่วิทยุที่ 2.4 GHz ที่ IEEE 802.11b ใช้อยู่นั้นยังมีอุปกรณ์อื่นๆ ร่วมใช้งานอยู่ด้วยหลายชนิด เช่น เตาไมโครเวฟ หรือ โทรศัพท์มือถือ ซึ่งหากมีอุปกรณ์เหล่านี้ทำงานอยู่ใกล้ๆ กับเครือข่าย IEEE802.11b ก็จะทำให้ความเร็วในการรับส่งข้อมูลช้าลง แต่จุดเด่นก็คือการใช้ความถี่คลื่นวิทยุที่ค่อนข้างต่ำ เพียง2.4 GHz นั้นทำให้ IEEE 802.11b มีระยะทางในการติดต่อระหว่างอุปกรณ์ค่อนข้างไกล ทำให้ชุดเครือข่ายไร้สายแบบ IEEE 802.11b ไม่จำเป็นต้องมีจุด รับส่งสัญญาณ หรือที่เรียกกันว่า Access Poin

ตารางที่ 2.1 อัตราการส่งข้อมูลจำเพาะ ของ IEEE 802.11b

Data Rate (Mbps) อัตราการส่งข้อมูล	Code Length ความยาวของรหัส	Modulation ชนิดของการมอดดูเลต	Symbol Rate (Mbps) อัตรา Symbol	Bits/ Symbol
1	11 (Barker Sequence) ลำดับของบาร์เกอร์	BPSK	1	1
2	11 (Barker Sequence) ลำดับของ บาร์เกอร์	QPSK	1	2
5.5	8(CCK)	QPSK	1.375	4
11	8(CCK)	QPSK	1.375	8

• มาตรฐาน IEEE802.11a

มาตรฐาน IEEE 802.11a นั้นเกิดขึ้นหลังการวางตลาดของมาตรฐาน IEEE 802.11b โดยผลิตภัณฑ์IEEE 802.11a มีจุดเด่นที่เหนือกว่า IEEE 802.11b ตรงที่ความเร็วในการรับส่งข้อมูลนั้นจะเร็วกว่า คือ ทำได้สูงสุดถึง 54 Mbps และเร็วกว่า IEEE 802.11b ในทุกระยะทาง (ความเร็วของเครือข่ายไร้สายทุกมาตรฐานจะลดลงเมื่อระยะทางมากขึ้น) โดยมีความถี่คลื่นวิทยุอยู่ที่ 5 GHz ซึ่งเป็นย่านความถี่วิทยุ ของ Unlicensed NationalInformation Infrastructure (U-NII) band มีความกว้างของความถี่ทั้งหมด 300 MHz โดยแบ่งเป็น 3 ระดับระดับละ 100 MHz คือ ต่ำ, ปานกลาง และสูง ซึ่งแต่ละระดับมีระดับมีการสามารถใช้งานและกำลังส่งแตกต่างกัน

- ย่านความถี่ระดับต่ำ (Low Band) ย่านความถี่ที่ทำงานจาก 5.15 ถึง 5.25 GHz กำลังส่งสูงสุด เท่ากับ 50 mW

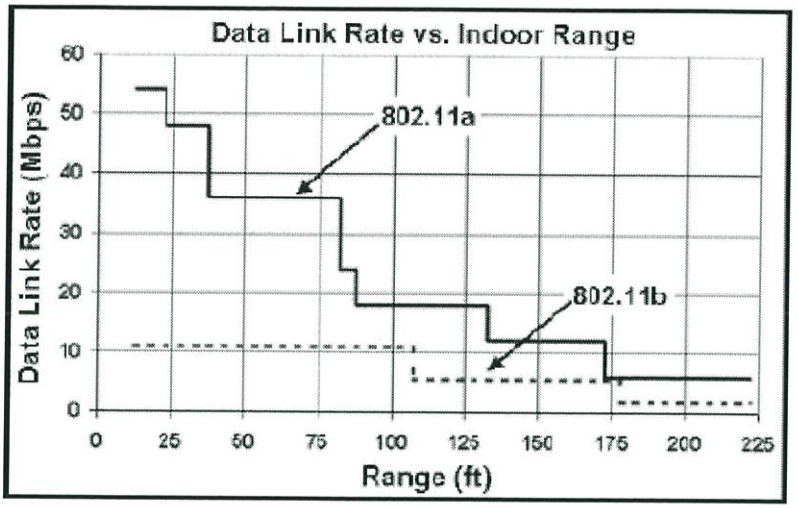
- ย่านความถี่ระดับปานกลาง (Middle Band) ย่านความถี่ที่ทำงานจาก 5.25 ถึง 5.35 GHz ด้วยกำลังส่งสูงสุด เท่ากับ 250 mW

- ย่านความถี่ระดับสูง (High Band) ย่านความถี่ที่ทำงานจาก 5.725 ถึง 5.825 GHz ด้วยกำลังส่งสูงสุด เท่ากับ 1000 mW

โดยกำลังส่งที่สูงของเครื่องรับ-ส่งสัญญาณของระบบเครือข่ายไร้สายและช่วงความถี่ 5.8 GHz จะทำให้สามารถส่งสัญญาณติดต่อกัน ระหว่างอาคารหนึ่ง กับอีกอาคารหนึ่งได้ ส่วนการใช้งานภายในอาคารจะใช้งานในย่านความถี่ระดับปานกลางและต่ำ ซึ่งในอเมริกาสามารถใช้งานได้ทั้ง 3 ย่านความถี่ แต่ปัญหาเรื่องของกฎหมายเกี่ยวกับคลื่นความถี่ระดับ 5 GHz ที่ในแถบยุโรปและประเทศญี่ปุ่นมีข้อกำหนด ค่อนข้างเคร่งครัดคือ ในยุโรปกำลังทำข้อตกลงร่วมกันระหว่าง IEEE และ European Telecommunications Standards Institute (ETSI) ส่วนในประเทศญี่ปุ่นอนุญาตให้ใช้ได้เฉพาะ ย่านความถี่ต่ำเท่านั้น

ดังนั้นการใช้งานในย่านความถี่ปานกลางและต่ำ จึงมีความกว้างของสัญญาณรวมกันเท่ากับ 200 MHz สามารถส่งข้อมูลได้ด้วยอัตราเร็วสูงสุดถึง 54 Mbps ได้สำเร็จ โดยใช้หลักการ ส่งสัญญาณความถี่ย่อยโดยอัตราเร็วต่ำๆ พร้อมๆกัน เมื่อนำทั้งหมดมารวมกัน ก็จะสามารถสร้างช่องสัญญาณที่มีอัตราเร็วสูงขึ้นได้ ตามที่ได้รับอนุญาตให้ใช้ช่วงความถี่ดังกล่าว สามารถแบ่งการใช้งานได้ ถึง 8 ช่องสัญญาณโดยไม่ทับซ้อนกัน แต่ละช่อง สัญญาณมีความกว้าง เท่ากับ 20 MHz ใช้การมอดดูเลชั่นแบบ OFDM (Orthogonal Frequency Division Multiplex) ในการส่งสัญญาณ ซึ่งเป็นเทคนิคการส่งสัญญาณแบบแยกส่งเป็นความถี่ย่อย ๆ (Narrow-Band Subcarriers) และมีความเป็นอิสระต่อกัน แต่ละความถี่ย่อยจะมีความกว้างเท่ากับ 300 KHz จำนวน 52 ช่อง สัญญาณความถี่ย่อย สัญญาณความถี่ย่อยจะทำการรับและส่งข้อมูลโดยส่งไปแบบขนาน ด้านรับสัญญาณจะได้รับข้อมูลทั้งหมดพร้อมกัน ซึ่งนั้นก็หมายความว่าข้อมูลที่ส่งจะมีขนาดใหญ่ และต้องการความต่อเนื่องในการส่ง สัญญาณ เพราะฉะนั้นเพื่อป้องกัน การสูญหายของข้อมูล (Data Loss Feature) จึงเพิ่ม Forward Error Correction (FEC) เข้าไปใน 802.11a ด้วย ซึ่งจะมีเฉพาะใน 802.11a เท่านั้น (ไม่พบใน 802.11b)

อัตราการส่งข้อมูลและระยะทางในการส่งข้อมูล มาตรฐาน 802.11a รองรับอัตราความเร็วของการส่งข้อมูล เท่ากับ 6, 9,12, 18, 24, 36, 48 และ 54 Mbps อัตราความเร็วจะลดลงเองอย่างอัตโนมัติขึ้นอยู่กับระยะทางระหว่าง Access point กับ เครื่องคอมพิวเตอร์ ลูกข่าย โดยที่ความเร็วสูงสุดที่ 54 Mbps นั้นใช้การมอดูเลชั่นสัญญาณความถี่ย่อย แบบ 64-level Quadrature Amplitude Modulation (64 QAM) คล้ายกันกับ 802.11b ที่ เครื่องลูกข่ายมาตรฐาน 802.11a จะมีอัตราเร็วลดลงเหมือนระยะทางจากAccess Point มากขึ้น แต่เมื่อนำมาเปรียบเทียบกันแล้ว 802.11a ยังมีความเร็วที่เหนือกว่าในทุกระยะทาง



รูปที่ 2.20 การเปรียบเทียบอัตราเร็วกับระยะทางระหว่าง มาตรฐาน 802.11a และ802.11b [5]

ตารางที่ 2.2 อัตราการส่งข้อมูลจำเพาะของ IEEE 802.11a

Data Rate(Mbps) อัตราการส่งข้อมูล	Code Rate (R) อัตราการส่งรหัส	Modulation ชนิดของการมอดดูเลชั่น	Code bits Per Subcarrier (NBPSC)	Code bits Per OFDM Symbol (NCBPS)	Data bits Per OFDM Symbol (NDBPS)
6	½	BPSK	1	48	24
9	¾	BPSK	1	48	36
12	½	QPSK	2	96	48
18	¾	QPSK	2	96	72
24	½	16-QAM	4	192	96
36	¾	16-QAM	4	192	144
48	2/3	16-QAM	6	288	192
54	3/4	16-QAM	6	288	216

• มาตรฐาน IEEE802.11g

มาตรฐานนี้เป็นมาตรฐานที่ความถี่ 2.4 GHz โดยสามารถรับส่งข้อมูลที่ความเร็ว 36 - 54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน 802.11b ซึ่ง 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ (ตามสภาพแวดล้อมของเครือข่ายที่ใช้งาน) มาตรฐานนี้เป็นที่ยอมรับจากผู้ใช้เป็นจำนวนมาก นอกจากที่กล่าวมาข้างต้นนี้มีบางผลิตภัณฑ์ใช้เทคโนโลยีเฉพาะตัวเข้ามาเสริมทำให้ความเร็วเพิ่มขึ้นจาก 54 Mbps เป็น 108 Mbps แต่ต้องทำงานร่วมกันเฉพาะอุปกรณ์ที่ผลิตจากบริษัทเดียวกันเท่านั้น ซึ่งความสามารถนี้เกิดจากชิป (Chip) กระจายสัญญาณของตัวอุปกรณ์ที่ผู้ผลิตบางรายสามารถเพิ่ม ประสิทธิภาพการรับส่งสัญญาณเป็น 2 เท่าของการรับส่งสัญญาณได้ แต่ปัญหาของการกระจาย สัญญาณนี้จะมีผลทำให้อุปกรณ์ไร้สายในมาตรฐาน 802.11b

มีประสิทธิภาพลดลงด้วยเช่นกัน ด้านล่างเป็นตารางมาตรฐาน IEEE802.11 ของเครือข่ายไร้สายได้ทำการเปรียบเทียบมาตรฐานต่างๆ เอาไว้ ดังตารางต่อไปนี้

ตารางที่ 2.3 เปรียบเทียบมาตรฐานของ Wireless Lan

มาตรฐาน	802.11	802.11a	802.11b	802.11g
Standard Approved	July 1997 กรกฎาคม 2540	September 1999 กันยายน 2542	September 1999 กันยายน 2542	Draft Stage Completion Expected in 2002.
Available Bandwidth	83.5 MHz	300 MHz	83.5 MHz	83.5 MHz
Unlicensed Frequencies of Operation	2.4-2.4835 GHz DSSS,FHSS	5.15-5.35 GHz,OFDM 5.725-5.825 GHz,OFDM	2.4-2.4835 GHz DSSS	2.4-2.4835 GHz DSSS, OFDM
Number of Non-Overlapping Channels	3 Indoor/Outdoor	4 Indoor (UNII1) 4 Indoor/Outdoor (UNII2) 4 Indoor/Outdoor (UNII3)	3 Indoor/Outdoor	3 Indoor/Outdoor
Data Rate Per Channel	1, 2 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps	1, 2, 5.5, 11 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps
Modulation Type	DQPSK (2 Mbps DSSS) DBPSK (1 Mbps DSSS) 4GFSK (2MbpsFHSS) 2GFSK (1MbpsFHSS)	BPSK (6, 9 Mbps) QPSK (12, 18 Mbps) 16-QAM (24, 36 Mbps) 64-QAM (48, 54 Mbps)	DQPSK/CCK (11, 5.5 Mbps) DQPSK (2 Mbps) DBPSK (1 Mbps)	OFDM/CCK (6,9,12,18,24,36,48 ,54) OFDM (6,9,12,18,24,36,48 ,54) DQPSK/CCK(22, 33, 11, 5.5) DQPSK (2 Mbps) DBPSK (1 Mbps)
Compatibility	802.11	Wiiii-Fi5	WiFi	WiFi at 11MbpsAnd Below

2.7.5 เทคโนโลยีในการส่งสัญญาณ Wireless Lan

โดยทั่วไปแล้วระบบเครือข่ายไร้สายจะใช้เทคโนโลยีในการส่งสัญญาณอยู่ 2 ประเภท คือ ประเภทที่ใช้ สัญญาณคลื่นความถี่วิทยุซึ่งแบ่งเป็น 2 แบบ คือ Narrow Band และ Spread Spectrum และประเภทที่ใช้ สัญญาณอินฟราเรด ในการติดต่อรับ - ส่ง ข้อมูล โดยมีรายละเอียดดังต่อไปนี้

2.7.5.1 Narrow band Technology

ระบบวิทยุแบบความถี่แคบ เป็นการรับส่ง สัญญาณคลื่นวิทยุบนความถี่เฉพาะโดยคลื่นความถี่ดังกล่าว เป็นที่รู้จักในชื่อของแถบความถี่ ISM (Industrial Scientific / Medical) ที่มีความถี่แบ่งเป็น 3 ช่วง ได้แก่ 902 MHz ถึง 928 MHz, 2.14 MHz ถึง 2.484 และ 5.725 MHz ถึง 5.850 MHz สัญญาณจะมีกำลังต่ำ (โดยทั่วไปประมาณ 1 มิลลิวัตต์) และใช้ในการรับ-ส่ง ข้อมูลระหว่างต้นทางกับ ปลายทางเพียง 1 คู่เท่านั้น และไม่สามารถส่งสัญญาณข้ามโหนดไปมาได้ การส่งข้อมูลแบบนี้ เปรียบได้กับคู่สาย โทรศัพท์ที่สามารถคุยได้เฉพาะต้นทางกับปลายทางแต่ ไม่สามารถคุยพร้อมกันได้ หลายๆ คน ข้อจำกัดของการใช้สัญญาณแบบนี้ คือจะต้องของอนุญาตจาก FCC (Federal Communication Committee) ซึ่งเป็นหน่วยงานที่กำหนดความถี่ในการใช้สัญญาณคลื่นวิทยุแบบ Narrow band นี้

2.7.5.2 Spread spectrum technology

ระบบเครือข่ายไร้สายส่วนใหญ่นิยมใช้เทคนิค Spread Spectrum Technology ซึ่งใช้ความถี่ที่กว้างกว่า Narrow Band Technology ซึ่ง Spread Spectrum ก็คือ วิธีการเปลี่ยนแปลงสัญญาณข้อมูลเพื่อให้ครอบคลุมพื้นที่ความถี่วิทยุมากเกินความจำเป็น แรกทีเดียว เทคนิคนี้ได้รับการพัฒนาขึ้นมาเพื่อใช้ในกิจการทางทหารซึ่ง ต้องการความเชื่อถือได้ในระดับสูงมาก ในระหว่างการรบ ข้าศึกอาจใช้อุปกรณ์อิเล็กทรอนิกส์ดักฟังสัญญาณเพื่อ ขโมยความลับหรือรบกวนการทำงาน แต่ในระบบนี้การส่งสัญญาณถูกส่งออกไปหลายความถี่พร้อมกันจึงทำให้การดักฟังเป็นไปได้ยากขึ้น รวมทั้งการรบกวนการสื่อสารก็ยากมากขึ้นด้วยเพราะจะต้องค้นหาคลื่นความถี่ทั้ง หมดให้ได้ โดยการส่งสัญญาณจะใช้แถบความถี่ ISM ที่ช่วงความถี่ ระหว่าง 902-928 MHz และ 2.4-2.484 GHz เทคนิค Spread Spectrum สามารถแบ่งได้ เป็น 2 แบบ คือ Direct Sequence และ Frequency – Hopping

2.7.5.3 Direct Sequence Spread Spectrum (DSSS)

Direct Sequence Spread Spectrum เป็นเทคนิคที่ยังใช้คลื่นพาหะที่ต้องระบุความถี่ที่ใช้ สามารถส่งข้อมูลได้มากกว่า แบบ Narrow Band ข้อมูลจะถูกกระจายให้ช่วงความถี่กว้างขึ้น (RF Bandwidth) ในรูปแบบ ของรหัสเฉพาะ รูปแบบของรหัสเฉพาะที่เป็นที่รู้จักกันดีคือ Pseudo-noise Sequence หรือ PN Sequence รูปแบบนี้จะใช้การเข้ารหัสในวิธีพิเศษ โดยการแปลงเลขฐานสองแต่ละบิทในข้อมูลดั้งเดิมที่จะส่งไปให้อยู่ในรูปแบบเลขฐานสองที่มีความยาวเพิ่มมากขึ้น ตัวอย่างเช่น ข้อมูลเลขฐานสอง 1 อาจจะถูกแปลงเป็น 0010010101 และข้อมูล 0 จะถูก

แปลงเป็น Inverse ของ 1 คือ 1101101010 แล้วข้อมูลที่แปลงแล้วเหล่านี้จะถูก ส่งไปพร้อมๆกันในลักษณะขนาน ซึ่งหากผู้รับสามารถจดจำรูปแบบการแปลงข้อมูลได้ก็จะถูกส่งไป โดยที่ สัญญาณรบกวนไม่สามารถทำให้ข้อมูลเสียหายไปได้ หรือหากรูปแบบที่ส่งไปเกิดผิดพลาดไปไม่ว่าจะด้วยสาเหตุ ใดก็ตาม ทางฝ่ายรับก็สามารถที่จะใช้เทคนิคในทางสถิติเพื่อกู้ข้อมูลที่ผิดพลาดไปให้กลับคืนมาได้ วิธีนี้จะใช้ใน มาตรฐาน IEEE802.11 และIEEE 802.11b ผู้ผลิตระบบเครือข่ายไร้สายส่วนใหญ่จะเลือกใช้วิธีการนี้เพราะว่า เป็นวิธีที่เหมาะมากกว่าวิธีอื่นในสภาพแวดล้อมที่มีการแทรกสอดรบกวนจากคลื่นวิทยุอื่นๆ อย่างรุนแรง นอกจาก นี้ยังเปิดโอกาสให้ผู้ใช้ตัดสินใจได้ว่าจะทำการจัดสรรแถบความถี่ในการส่งข้อมูลอย่างไรบ้าง เช่น อาจจัดแบ่ง แถบความถี่เป็นช่วงย่อยหลายช่วงเพื่อใช้ส่งข่าวสารหลายชิ้นไปพร้อมกัน

2.7.5.4 Frequency – Hopping Spread Spectrum (FHSS)

การส่งสัญญาณรูปแบบนี้จะใช้ความถี่แคบพาหะเพียงความถี่เดียว (Narrow Band) และจะเปลี่ยนแปลง ความถี่(กระโดด)ไปมาอย่างต่อเนื่อง ในลักษณะหรือรูปแบบที่เป็นที่เข้าใจตรงกันระหว่างเครื่องส่งกับเครื่องรับสามารถทำงานประสานกันได้แล้ววิธีการส่งแบบนี้ป้องกันสัญญาณรบกวนที่เกิดจากความถี่ข้างเคียงได้เป็นอย่างดี เพราะว่าความถี่จะมีการเปลี่ยนแปลงตลอดเวลา โดยการส่งและรับแต่ละครั้งที่ส่วนหัวของ Packet ข้อมูลจะบอก รับก็สามารถที่จะปรับเปลี่ยนไปได้ตลอดเวลาอันจะทำให้เกิดความปลอดภัยของข้อมูลสูงมากขึ้น ผู้ผลิตระบบเครือข่ายเฉพาะที่ไร้สาย แบบ Frequency Hopping ให้ความเห็นว่าการส่งข้อมูลวิธีนี้สามารถส่งข้อมูลไปพร้อมๆกันหลายช่องสัญญาณ ได้ด้วยการกำหนดให้มีรูปแบบของการเปลี่ยนแปลงหลายๆรูปแบบทำงานไปพร้อมกัน ซึ่งจะทำสามารถใช้ประโยชน์แถบความถี่ได้ดีกว่าและทำให้เครือข่ายมีประสิทธิภาพสูงกว่า ในการตัดสินใจเลือกใช้วิธีใดวิธีหนึ่งนั้น การนำไปใช้งานจะเป็นตัวกำหนดว่า ถ้าคำนึงถึงปัญหาทาง ด้านประสิทธิภาพและคลื่นรบกวนก็ควรใช้วิธี DSSS ถ้าต้องการใช้อะแดปเตอร์ไร้สายขนาดเล็กและราคาไม่แพง สำหรับเครื่องโน๊ตบุ๊ค หรือ เครื่อง PDA ก็ควรเลือกแบบ FHSS

2.7.5.5 Orthogonal frequency division multiplex (OFDM)

เทคนิคนี้ถูกนำมาใช้ในเพื่อเพิ่มความเร็วในการส่งข้อมูลในมาตรฐาน ใหม่ๆ ของระบบเครือข่ายไร้สาย คือ IEEE 802.11a และ 802.11g การส่งสัญญาณคลื่นวิทยุแบบนี้ เป็นการมัลติเพล็กสัญญาณ โดยช่องสัญญาณ ความถี่จะถูกแบ่งออกเป็นความถี่พาหะย่อย (Subcarrier) หลายๆความถี่ โดยแต่ละความถี่พาหะย่อยจะตั้งฉาก ซึ่งกันและกัน ทำให้มันเป็นอิสระต่อกัน ความถี่ที่คลื่นพาหะที่ตั้งฉากกันนั้นทำให้ไม่มีปัญหาการซ้อนทับกันของ สัญญาณที่อยู่ติดกัน OFDM เป็นเทคนิคการมัลติเพล็กโดยการแบ่งความถี่ เมื่อช่องความถี่ถูกแบ่งออกเป็นขนาดเล็กๆ N ช่องแต่ละช่องมีขนาดเท่ากับขนาดของสัญลักษณ์ (Bit Rate) ดิจิตอล ทางด้านส่งจะมีสัญญาณดิจิตอลความเร็วสูงที่ถูกแบ่งออกเป็นกลุ่มข้อมูลย่อยๆ ที่มีความถี่ต่ำกว่า จะถูกมอดูเลตกับสัญญาณพาหะย่อย 1 สัญญาณ และนำสัญญาณ ทั้งหมดส่งขนานกันออกไป รูปแบบในการมอดูเลตสัญญาณพาหะย่อยที่นิยมทั่วไปได้แก่ QAM, 16 QAM หรือ 64 QAM เป็นต้น ในOFDM กลุ่มของข้อมูลจะถูก

แปลงให้อยู่ในรูปขนานกัน โดยการมอดูเลตกับสัญญาณพาหะ ย่อย ดังนั้น จะกลายมาเป็นสัญญาณบนแกนความถี่ ซึ่งการแปลงสัญญาณกลับให้อยู่บนแกนเวลาอีกครั้งโดยการ แปลงกลับฟาสท์ฟูเรียร์ (IFFT) จากนั้นจะสัญลักษณ์บนแกนเวลาจะถูกมัลติเพล็กเข้าด้วยกันให้เป็นอนุกรมของ สัญญาณแล้วจึงส่งสัญญาณออกไปทางเสาอากาศ หลังจากการมอดูเลตแบบ OFDM จะมีการสอดแทรกช่วงแถบป้องกันแคบๆ เพื่อลดสัญญาณรบกวนระหว่างสัญลักษณ์ (Inter Symbol Interference: ISI) ที่เกิดจากสัญญาณหลายเส้นทาง (Multi-Path) เรียกแถบ ป้องกันแคบๆนี้ว่า การเสริมไซคริก (Cyclic prefix) ส่วนในเครื่องรับจะดำเนินกระบวนการตรงข้ามกับเครื่องส่งในเครื่องรับจะใช้การแปลงฟาสท์ฟูเรียร์แปลงสัญญาณที่อยู่บนแกนเวลาไปเป็นแถบความถี่สมมูลย์ ข้อดีของ OFDM คือสามารถใช้งานแถบความถี่ในระบบที่เคยใช้สัญญาณพาหะเดี่ยวได้อย่างเต็มประสิทธิภาพ (Spectral Efficiency) สามารถป้องกันผลกระทบจากการเคลื่อนที่ของสัญญาณหลายเส้นทาง (Immunity to Muti-Path) และมีความไวต่ำต่อการเลื่อนหายไปของความถี่ที่เลือก (Less Sensitivity to Frequency Selective Fading)

2.7.5.6 Infrared Technology

ลำแสงอินฟราเรด (Infrared : IR) เป็นส่วนหนึ่งของสเปกตรัมแม่เหล็กไฟฟ้าอยู่ในย่านความถี่ของแสงที่อยู่ต่ำกว่าแสงสีแดงที่ตาของคนจะไม่สามารถมองเห็นลำแสงที่มีความถี่ระดับนี้ เป็นลำแสงอีกชนิดหนึ่งที่ถูกนำมาใช้เพื่อการสื่อสารที่ใช้ในระยะใกล้ ตัวอย่างของการใช้ลำแสงอินฟราเรดที่พบได้ในชีวิตประจำวันได้แก่ อุปกรณ์ควบคุมระยะไกลแบบไร้สาย (Wireless Remote Control) ที่ใช้ควบคุมเครื่องรับโทรทัศน์ เครื่องเล่นวีดีโอ เครื่องเสียงและอุปกรณ์ไฟฟ้าต่างๆ ในระบบคอมพิวเตอร์ เครื่องคอมพิวเตอร์และอุปกรณ์บางชนิดได้ใช้ลำแสง อินฟราเรดเพื่อการติดต่อสื่อสารระหว่างกันแบบจุดต่อจุด เช่นคอมพิวเตอร์โน๊ตบุ๊ค เครื่องคอมพิวเตอร์มือถือ (Hand Held) หรือเครื่องขนาดฝ่ามือ(Palmtop) จะมีพอร์ตอินฟราเรดเพื่อรับส่งข้อมูลออกไปยังเครื่องพิมพ์ซึ่ง เครื่องพิมพ์แบบเลเซอร์สำหรับสำนักงานบางรุ่นจะรับข้อมูลผ่านทางพอร์ตอินฟราเรด คุณสมบัติเด่นของคลื่นอินฟราเรดและคลื่นสั้น คือเดินทางเป็นแนวตรง ราคาถูก และง่ายต่อการผลิตใช้งานแต่คลื่นประเภทนี้ไม่สามารถเดินทางผ่านวัตถุหรือสิ่งกีดขวางได้ ซึ่งเป็นข้อดีคือสามารถนำอุปกรณ์ที่ใช้คลื่นอินฟราเรดมาใช้ในห้องทำงานที่อยู่ติดกันได้แม้ว่าอุปกรณ์ทั้งสองชิ้นนั้นจะใช้ความถี่เดียวกัน ยิ่งกว่านั้นอุปกรณ์ที่ใช้คลื่นอินฟราเรดยังปลอดภัยต่อการถูกลักลอบดักสัญญาณด้วย คุณสมบัติเหล่านี้ทำให้คลื่นอินฟราเรดสามารถนำมาใช้ในการสื่อสารในระบบเครือข่ายเฉพาะบริเวณได้เป็นอย่างดี

หลักการทำงานของอินฟราเรดมี ดังนี้

1. จัดตำแหน่งในการพิมพ์ไฟล์จากคอมพิวเตอร์โน๊ตบุ๊ค ให้วางอุปกรณ์นั้น 3 ฟุตจากเครื่องพิมพ์ที่เหมาะสมกับ IR ชี้ที่พอร์ต IR (หรือที่เรียกว่าโฟโตไดโอด) ตรงไปยังโฟโตไดโอดของเครื่องพิมพ์

2. ส่งพัลส์ของแสงอินฟราเรดจะถูกส่งไปกลับระหว่างอุปกรณ์สองตัวเพื่อขนถ่ายแพ็กเกตของข้อมูลที่ประกอบกันเป็นแพ็กเกตจะถูกสื่อสารด้วยพัลส์เปิด/ปิดของแสงอินฟราเรดโดยพัลส์จะถูกอ่านในรปของรหัสไบนารี่

3. รับโฟโตไดโอดจะรับแพ็กเกต ซึ่งจะถูกแปรกลับไปเป็นข้อมูลอีกครั้ง เครื่องพิมพ์หรือพีซีในด้านรับจะประมวลผลข้อมูลที่ได้มาจากการเชื่อมต่อเครือข่ายที่ใช้สายเคเบิล

4. การขัดขวางถ้ามีวัตถุมาขัดขวางลำของพัลส์ของอินฟราเรดขณะ ที่ข้อมูลกำลังถูกส่งสัญญาณจะถูกบล็อก อย่างไรก็ตามอุปกรณ์ด้านส่งจะรับรู้ข้อผิดพลาดและทำการส่งข้อมูลที่ขาดหายไปใหม่

ตารางที่ 2.4 เปรียบเทียบคุณสมบัติของคลื่นความถี่วิทยุและแสงอินฟราเรด

คุณสมบัติ	Spread Spectrum	อินฟราเรด
Frequency ความถี่	902 MHz to 928 MHz 2.4 GHz to 2.4385 GHz 5.725 GHz to 5.825 GHz	3×10^{14} Hz
Maximum coverage ครอบคลุมพื้นที่ส่วนใหญ่	105 to 800 feet, or up 50,000 square feet	30 to 80 feet
Line of sight required ความต้องการเส้นทาง	No	Yes
Transmit power กำลังส่ง	Less than 1 w	N/A
License required ลิขสิทธิ์	No	No
Inter-building use การใช้ภายในอาคาร	Possible with antenna	Possible
Rated speed (% of 10 Mbps wire) อัตราเร็ว	20% to 50%	50% to 100%

2.7.6 อุปกรณ์สำหรับ Wireless Lan

2.7.6.1 Access Point ทำหน้าที่เป็นสะพานเชื่อมระหว่าง Ethernet ของ Network แบบมีสายกับเครื่อง Client ที่เชื่อมต่อแบบไร้สาย (อาจเรียกเป็น Hub ไร้สาย หรือ Switch ไร้สาย) สามารถ รับ-ส่ง ข้อมูลสำหรับผู้ใช้ได้หลายเครื่องพร้อมๆกัน

รูปที่ 2.21 Access point ที่มีการกระจายสัญญาณแบบ Multiple-Input Multiple-Output [3]

2.7.6.2 Wireless LAN card ทำหน้าที่ในการ แปลงข้อมูล ดิจิตอล ที่ได้จากการประมวลผลของเครื่องคอมพิวเตอร์ให้เป็นคลื่นวิทยุแล้วส่งผ่านสายอากาศให้กระจายออกไป และทำหน้าที่ในการรับเอาคลื่นวิทยุที่แพร่กระจายแปลงเป็น ข้อมูลดิจิตอล ส่งให้เครื่องคอมพิวเตอร์ประมวลผล Wireless LAN ที่ผลิตออกมาจำหน่าย มีหลายรูปแบบแบ่งตามลักษณะช่องเชื่อมต่อคอมพิวเตอร์ได้ดังนี้

- แลนการ์ดแบบ PCI
- แลนการ์ดแบบ PCMCIA
- แลนการ์ดแบบ USB
- แลนการ์ดแบบ Compact Flash (CF)

รูปที่ 2.22 Wireless Lan Card แบบ USB [3]

2.7.7 ข้อคำนึงในการเลือกซื้ออุปกรณ์ที่ใช้งานในระบบเครือข่ายไร้สาย

- การเลือกมาตรฐานให้เหมาะสมสำหรับการใช้งานโดยในปัจจุบันมาตรฐานที่นิยมใช้กันมากคือมาตรฐาน IEEE802.11g ซึ่งรองรับอัตราความเร็วสูงสุดในระดับ 54 Mbps โดยเป็นความเร็วที่เพียงพอสำหรับการใช้งานโดยทั่วๆ ไปในปัจจุบันได้อย่างดีแล้วนอกจากนั้นยังสนับสนุนการทำงานร่วมกันกับมาตรฐานเดิมอย่าง IEEE802.11b ได้ แต่ในขณะนี้ผู้ผลิตหลายๆ รายต่างแข่งขันกันผลิตผลิตภัณฑ์ที่สนับสนุนเทคโนโลยี MIMO ออกมามากขึ้น โดยเทคโนโลยี MIMO นี้เป็นเทคโนโลยีที่ใช้เทคนิคการใช้ตัวส่งตัวรับสัญญาณหลายตัวซึ่งทำให้การถ่ายโอนข้อมูลสามารถทำได้เร็วขึ้นด้วยการใช้ประโยชน์จาก Multipath ข้อมูลหลายชุดจึงถูกส่งและรับได้ในเวลาเดียวกันจึงเป็นที่คาดหมายกันว่าในอนาคตเครือข่ายไร้สายที่มีประสิทธิภาพการใช้งานที่มากกว่า ให้แบนด์วิดท์สูงและมีรัศมีการทำงานที่ดีกว่านั้นจะเข้ามาทดแทนมาตรฐาน IEEE 802.1g เดิม แต่อย่างไรก็ตามผลิตภัณฑ์ที่จะใช้งานคุณสมบัติเหล่านี้ได้อย่างเต็มที่จะต้องเป็นอุปกรณ์จากชุดเดียวกันซึ่งปัจจุบันอุปกรณ์เหล่านี้ยังคงมีราคาแพงอยู่มาก ดังนั้นการเลือกใช้อุปกรณ์สำหรับมาตรฐาน IEEE802.11g จึงยังคงเป็นทางเลือกที่ดีที่สุดในขณะนี้

- การเลือกระบบอินเตอร์เฟซที่เหมาะสม สำหรับการ์ดอีเทอร์เน็ตไร้สายในปัจจุบันนั้นมีหลายชนิดให้เลือกใช้เช่นเดียวกัน ส่วนเครื่องคอมพิวเตอร์แบบโน้ตบุ๊กก็มีการผนวกรวมคุณสมบัติแบบไร้สายมาพร้อมกับตัวเครื่องแล้ว หากเครื่องคอมพิวเตอร์แบบโน้ตบุ๊กไม่มีคุณสมบัติที่ใช้งานกับระบบเครือข่ายไร้สายในตัวเครื่องก็สามารถใช้ Wireless PCMCIA Card ติดตั้งเข้าไปในตัวเครื่อง หรือถ้าต้องการใช้เครื่องคอมพิวเตอร์แบบพีซีร่วมกับระบบเครือข่ายไร้สายก็ควรเลือกใช้การ์ดแบบ USB Adapter ซึ่งราคาอาจจะค่อนข้างสูงแต่สามารถใช้งานได้ความคุ้มค่าและหลากหลายกว่าสำหรับการใช้งานเครื่องคอมพิวเตอร์แบบพีซีกับระบบเครือข่ายไร้สายเพียงอย่างเดียวก็ใช้

อินเทอร์เฟซแบบ PCI Card ได้ซึ่งส่วนใหญ่จะมีสายสัญญาณและเสาอากาศที่ตั้งบนที่สูงเพื่อเพิ่มประสิทธิภาพของการสื่อสารได้

- การเลือกผลิตภัณฑ์เชื่อมโยงสัญญาณระหว่างกัน (Access Point / Wireless Router) เพราะนอกจากอุปกรณ์เหล่านี้จะสนับสนุนการทำงานในแบบ Ad-Hoc หรือ Peer-to-Peer แล้ว ระบบเครือข่ายไร้สายก็ยังสามารถใช้ Access Point เป็นจุดเชื่อมต่อสัญญาณกับเครือข่ายใช้สายเพื่อการแชร์การใช้ทรัพยากรร่วมกันได้อย่างมีประสิทธิภาพและตอบสนองความต้องการได้มากกว่าแบบ Insfrastructure โดยถ้ายังไม่มีการเชื่อมต่ออินเทอร์เน็ตหรือติดตั้งระบบเครือข่ายมาก่อนก็ควรจะเลือกใช้อุปกรณ์อย่าง Wireless Router ที่มีคุณสมบัติในแบบ All-in-One เพราะสามารถเป็นทั้ง Router Switch และ Access Point ในเครื่องเดียวซึ่งจะให้ความคุ้มค่ามากกว่าหรือหากมีการใช้งานเครือข่ายใช้สายและไร้สายอยู่ก่อนแต่ต้องการเพิ่มประสิทธิภาพของการใช้งาน ดังนั้นควรเลือกใช้ Access Point ที่สนับสนุนโหมดการทำงานแบบ Bridge และ Repeater ร่วมด้วย

- การใช้งานระบบรักษาความปลอดภัย สิ่งที่ต้องคำนึงเป็นพิเศษในการเลือกซื้อผลิตภัณฑ์ระบบเครือข่ายไร้สายคือต้องให้ความสนใจในการเข้ารหัสข้อมูลเพราะการสื่อสารไร้สายนั้นเป็นการติดต่อสื่อสารด้วยการใช้คลื่นวิทยุที่แพร่ไปตามบรรยากาศ ทั้งนี้ก็เพื่อป้องกันการดักจับสัญญาณจากผู้ไม่ประสงค์ดี ดังนั้นการเลือกซื้อผลิตภัณฑ์ไร้สายจึงต้องคำนึงถึงฟังก์ชันการเข้ารหัสที่ใช้ ซึ่งเทคนิคที่ใช้งานโดยทั่วๆ ไปสำหรับผู้ใช้ตามบ้าน Wired Equivalent Privacy หรือ WEP ขนาด 64/128-bit ร่วมกับ MAC Address Filtering นั้นก็เพียงพอแล้ว แต่สำหรับการใช้งานภายในองค์กรนั้นควรใช้เทคนิคการตรวจสอบและกำหนดสิทธิ์การใช้งานที่สูงกว่าโดยเลือกใช้ WPA (WiFi Protected Privacy) ซึ่งใช้คีย์การเข้ารหัสที่น่าเชื่อถือร่วมกับเทคนิคการตรวจสอบและการกำหนดสิทธิ์ในแบบ 2 ฝั่งหรืออาจจะใช้ระบบรักษาความปลอดภัยแบบอื่นๆ เช่น RADIUS ร่วมด้วยก็ได้

- สำหรับเสาอากาศของการ์ดไร้สายนั้นถ้าเป็นการ์ดแบบ PCMCIA และแบบ USB จะเป็นเสาอากาศ Built-in มาพร้อมตัวการ์ด ส่วนการ์ดแบบ PCI นั้นจะเป็นเสาอากาศแบบ Reverse-SMA Connector ซึ่งสามารถถอดออกได้โดยทั่วไปจะเป็นทั้งแบบเสาเดี่ยวที่หมุนเข้ากับตัวการ์ดและอีกแบบคือมีสายนำสัญญาณต่อเชื่อมกับเสาที่ตั้งบนพื้นหรือยึดติดกับผนังได้ สำหรับการเลือกซื้อนั้นควรเลือกซื้อเสาอากาศที่มีสายนำสัญญาณต่อเชื่อมกับเสาที่ตั้งบนพื้นหรือยึดติดกับผนังเนื่องจากให้ความยืดหยุ่นในการติดตั้งมากกว่าเพราะสามารถติดตั้งบนที่สูงๆ ได้ ส่วนอุปกรณ์เชื่อมโยงสัญญาณระหว่างกัน อาทิ Access Point หรือ Wireless Router นั้นจะมีเสานำสัญญาณทั้งในแบบเสาเดี่ยวและ 2 เสาซึ่งการเลือกซื้อนั้นควรเลือกซื้อแบบ 2 เสา เนื่องจากให้ประสิทธิภาพในการรับส่งสัญญาณที่ดีกว่าโดยลักษณะของเสานั้นจะมีทั้งในแบบที่ยึดติดกับเข้ากับตัวอุปกรณ์ซึ่งส่วนใหญ่จะพบเห็นในรุ่นที่ออกแบบมาสำหรับผู้ใช้งานตามบ้านและอีกแบบเป็นเสาที่สามารถถอดเปลี่ยนได้ซึ่งหัวเชื่อมต่อนั้นจะเป็นทั้งแบบ Reverse-SMA Conector SMA Conector และแบบ T-Connector ซึ่งถ้ามีความจำเป็นที่จะต้องเปลี่ยนเสาอากาศควรจะเลือกซื้อจากทางผู้ผลิตรายเดียวกันเพื่อให้แน่ใจว่าจะไม่ซื้อหัวเชื่อมต่อผิดประเภท สำหรับชนิดของเสาอากาศที่มีจำหน่ายจะมี

2 ชนิดหลักก็คือ แบบ Omni-Direction Antenna ซึ่งเป็นเสาที่ทุกผู้ผลิตให้มากับตัวผลิตภัณฑ์แล้ว โดยคุณสมบัติของเสาประเภทนี้คือ การรับและส่งสัญญาณในแบบรอบทิศทางในลักษณะเป็นวงกลม ทำให้การกระจายสัญญาณนั้นมีรัศมีโดยรอบครอบคลุมพื้นที่ หากต้องการใช้งานที่มีลักษณะรับส่งสัญญาณเป็นเส้นตรงเพื่อให้ได้ประสิทธิภาพการรับส่งและระยะทางตามต้องการก็ควรใช้เสาแบบ Direction Antenna ซึ่งนิยมใช้งานกับผลิตภัณฑ์ประเภท Wireless Bridge สำหรับการสื่อสารในแบบ Point-to-Point ส่วนการเพิ่มระยะทางการเชื่อมต่อให้ได้ไกลมากยิ่งขึ้น ก็สามารถเลือกใช้เสาอากาศ High Gain ที่มีการขยายสัญญาณสูงกว่าเสาอากาศที่ทางผู้ผลิตให้มากับตัวอุปกรณ์โดยมีให้เลือกใช้หลายแบบทั้งในแบบที่มีค่า Gain 5 Gain 8 Gain 12 Gain 14 หรือ Gain ที่สูงกว่าได้

- กำลังส่งที่ปรับได้ สำหรับการใช้งานผลิตภัณฑ์ไร้สายนั้นการปรับกำลังส่งสัญญาณได้เป็นคุณสมบัติหนึ่งของผลิตภัณฑ์โดยกำลังส่งสูงสุดจะไม่เกิน 100mW หรือ 20dBm ผู้ผลิตบางรายจะมีผลิตภัณฑ์ที่สนับสนุนกำลังสูงสุดนี้ ซึ่งค่ากำลังส่งที่มากก็แสดงว่าสามารถที่จะแพร่สัญญาณไปในระยะทางที่ไกลหรือให้รัศมีที่มากขึ้น แต่ก็สามารถปรับกำลังส่งให้ลดต่ำลงเพื่อให้เหมาะสมกับความต้องการได้ โดยเฉพาะอย่างยิ่งการใช้งานภายในองค์กรที่จะต้องใช้กำลังส่งให้เหมาะสมกับพื้นที่ เนื่องจากกำลังส่งสูงๆ อาจจะไปรบกวนสำนักงานข้างเคียงและอาจถูกลักลอบใช้งานระบบเครือข่ายไร้สายก็เป็นได้

- ความเข้ากันได้ของผลิตภัณฑ์ไร้สายเพราะการใช้งานระบบเครือข่ายไร้สายให้ได้ประสิทธิภาพสูงสุดนั้นขึ้นอยู่กับความเข้ากันได้ของผลิตภัณฑ์ไร้สายด้วย หากผลิตภัณฑ์ไร้สายของผู้ผลิตแต่ละรายไม่สามารถทำงานเข้ากันได้กับผู้ผลิตรายอื่นก็จะทำให้การใช้งานเครือข่ายไร้สายด้อยประสิทธิภาพลงไป ดังนั้นเพื่อให้การใช้งานระบบเครือข่ายไร้สายได้ประสิทธิภาพและความคุ้มค่าที่สุดควรเลือกใช้ผลิตภัณฑ์ทั้งหมดจากผู้ผลิตรายเดียวกัน ซีรีส์เดียวกันหรือถ้าเลือกใช้ผลิตภัณฑ์ต่างผู้ผลิตก็ควรตรวจสอบแน่ใจว่าเลือกใช้ชิปเซ็ตซึ่งสนับสนุนเทคโนโลยีเดียวกันและก่อนการเลือกซื้อควรตรวจสอบความเข้ากันได้ของผู้ผลิตแต่ละรายโดยสังเกตได้จากตราสัญลักษณ์ที่ผ่านการรับรองจาก WiFi ก่อน

2.8 ทฤษฎีการเคลื่อนที่

การเคลื่อนที่ คือการเปลี่ยนตำแหน่งของวัตถุในช่วงเวลาหนึ่ง ซึ่งวัดโดยผู้สังเกตที่เป็นส่วนหนึ่งของกรอบอ้างอิง เมื่อปลายคริสต์ศตวรรษที่ 19 เซอร์ไอแซก นิวตัน ได้เสนอกฎการเคลื่อนที่ของนิวตันในหนังสือ Principia ของเขา ซึ่งต่อมาได้กลายเป็นกฎพื้นฐานของฟิสิกส์ดั้งเดิม

2.8.1 กฎการเคลื่อนที่ของนิวตัน

มนุษย์มีความพยายามที่จะอธิบายความสัมพันธ์ของแรงกระทำและการเคลื่อนที่ของวัตถุมาเป็นระยะเวลานานมาแล้ว อย่างไรก็ตามการอธิบายกฎการเคลื่อนที่ด้วยสมการทางคณิตศาสตร์หรือหลักทางกายภาพ ได้เริ่มมีขึ้นอย่างเป็นรูปธรรมในช่วง ศตวรรษที่ 16 โดย กาลิเลโอได้ทำการทดลองการเคลื่อนที่ของลูกตุ้มและการตกอย่างอิสระของวัตถุ ซึ่งก็ได้ผลสรุปเป็นกฎคร่าวๆ ที่เชื่อมโยงผลของแรงที่กระทำที่มีต่อการเคลื่อนที่ของวัตถุ อย่างไรก็ตามกฎการเคลื่อนที่ที่สามารถใช้ได้กับวัตถุในสภาพทั่วไปนั้น ได้เป็นที่ยอมรับกันว่าได้กำหนดขึ้นโดย Sir Isaac Newton ซึ่งได้นำเสนอกฎการเคลื่อนที่ของวัตถุในปี ค.ศ. 1687 โดยกฎการเคลื่อนที่ทั้งสามข้อมีดังนี้

กฎข้อที่ 1 วัตถุถ้าหากว่ามีสภาพหยุดนิ่งหรือเคลื่อนที่เป็นเส้นตรงด้วยความเร็วคงที่ มันยังจะคงสภาพเช่นนี้ต่อไป หากไม่มีแรงที่ไม่สมดุลจากภายนอกมากระทำ

กฎข้อที่ 2 ถ้าหากมีแรงที่ไม่สมดุลจากภายนอกมากระทำต่อวัตถุ แรงที่ไม่สมดุลนั้นจะเท่ากับอัตราการเปลี่ยนแปลงโมเมนต์ตัมเชิงเส้นของวัตถุ

กฎข้อที่ 3 ทุกแรงกริยาที่กระทำ จะมีแรงปฏิกิริยาที่มีขนาดที่เท่ากันแต่มีทิศทางตรงกันข้ามกระทำตอบเสมอ

กฎข้อที่สองของนิวตันกล่าวถึงความสัมพันธ์ของโมเมนต์ตัม ซึ่งคือผลคูณของมวลของวัตถุกับความเร็วของวัตถุ หรือ

$$\text{Momentum} = \text{Mass} \times \text{Velocity} \quad (P=mv) \tag{2.1}$$

โดยนิวตันกำหนดว่ าถ้ามีแรงมากระทำ ขนาดและทิศทางของแรงนั้น จะมีผลต่อการเปลี่ยนแปลงขนาดและทิศทางของโมเมนต์ตัม นั่นคือ

$$\vec{F} = \frac{d(m\vec{v})}{dt} \tag{2.2}$$

และในเมื่อในกลศาสตร์ มักจะพิจารณาระบบที่มีมวลคงที่ ดังนั้นจะได้ว่า

$$\vec{F} = m\frac{d(\vec{v})}{dt} = m\vec{a} \tag{2.3}$$

2.8.2 ระยะทาง ความเร็ว ความเร่ง

ระยะทาง (Distance) หมายถึงตัวเลขที่อธิบายว่า วัตถุแต่ละอย่างอยู่ห่างกันเท่าไรในช่วงเวลาหนึ่ง ในทางฟิสิกส์ ระยะทางอาจหมายถึงความยาวทางกายภาพ ระยะเวลา หรือการประมาณค่าบนสิ่งที่พิจารณาสองอย่าง ส่วนทางคณิตศาสตร์จะพิจารณาอย่างเฉพาะเจาะจงมากกว่าโดยทั่วไปแล้ว ระยะทางจาก A ไป B มีความหมายเหมือนกับ ระยะทางระหว่าง A กับ B

การกระจัด (Displacement) คือ เส้นตรงที่เชื่อมโยงระหว่างจุดเริ่มต้น และจุดสุดท้ายของการเคลื่อนที่เป็นปริมาณเวกเตอร์ คือ ต้องคำนึงถึงทิศทางด้วย มีหน่วยเป็นเมตร โดยทั่วไปเขียนแบบเว็กเตอร์เป็น S มีหมวกข้างบน

อัตราเร็ว (Speed) คือ ระยะทางที่วัตถุเคลื่อนที่ได้ในหนึ่งหน่วยเวลา จัดเป็นเปริมาณสเกลลาร์ หน่วยในระบบเอสไอ มีหน่วยเป็น เมตร/วินาที

ความเร็ว (Velocity) คือ ขนาดของการกระจัดที่วัตถุเคลื่อนที่ได้ในหนึ่งหน่วยเวลา จัดเป็นปริมาณเวกเตอร์ มีหน่วยเดียวกับอัตราเร็วคือ เมตร/วินาที เวกเตอร์ความเร็วขณะหนึ่ง v ของวัตถุที่มีตำแหน่ง d (t) ณ เวลา t และตำแหน่ง x (t + Δt) ณ เวลา t + Δt สามารถคำนวณได้จากอนุพันธ์ของตำแหน่ง

$$\vec{v} = \frac{d\vec{x}}{dt} \qquad (2.4)$$

ความเร่ง (Acceleration) คือ ความเร็วที่เปลี่ยนไปในหนึ่งหน่วยเวลาเป็นปริมาณเวกเตอร์ หรืออัตราการเปลี่ยนความเร็ว มีหน่วยเป็น ความยาว/เวลากำลังสอง ความเร่งมีสามชนิด คือ

1. อัตราเร่งที่เป็นเส้นตรง (Linear Acceleration) คือ การเปลี่ยนแปลงความเร็วที่ไม่มีการเปลี่ยนทิศทาง เช่น การวิ่งขึ้นจากสนามบินของเครื่องบิน การลงสนามบินของเครื่องบินหรือการดีดตัวออกจากเครื่องบินของเก้าอี้ดีด เป็นต้น

2. อัตราเร่งที่เกิดจากการเลี้ยวในแนวโค้ง (Radial Acceleration) คือ การเปลี่ยนทิศทางโดยมีความเร็วคงที่ ทำให้เกิดแรงหนีศูนย์กลาง (Centrifugal Force) เช่น การเลี้ยวด้วยความเร็วคงที่ของเครื่องบิน เป็นต้น

3. อัตราเร่งที่เกิดจากการเปลี่ยนแปลงความเร็วและทิศทางการเลี้ยว (Angular Acceleration) คือ การเลี้ยวที่เปลี่ยนแปลงทั้งความเร็วและทิศทาง แรง G ที่เกิดขึ้นเป็นผลรวมของอัตราเร่งชนิดแรก และชนิดที่สองรวมกันการบินในท่าวงกลมทางตั้ง (Loop) เป็นต้น อัตราการเปลี่ยนแปลงความเร็วของวัตถุในช่วงเวลาหนึ่ง และกำหนดโดยสมการนี้

$$\vec{a} = \frac{d\vec{v}}{dt} \qquad (2.5)$$

เมื่อ $\vec{a}$ คือ เวกเตอร์ความเร่ง

$\vec{v}$ คือ เวกเตอร์ความเร็ว ในหน่วย m/s

t คือ เวลา ในหน่วยวินาที

จากสมการนี้ $\vec{a}$ จะมีหน่วยเป็น m/s² (อ่านว่า "เมตรต่อวินาทียบกำลังสอง")

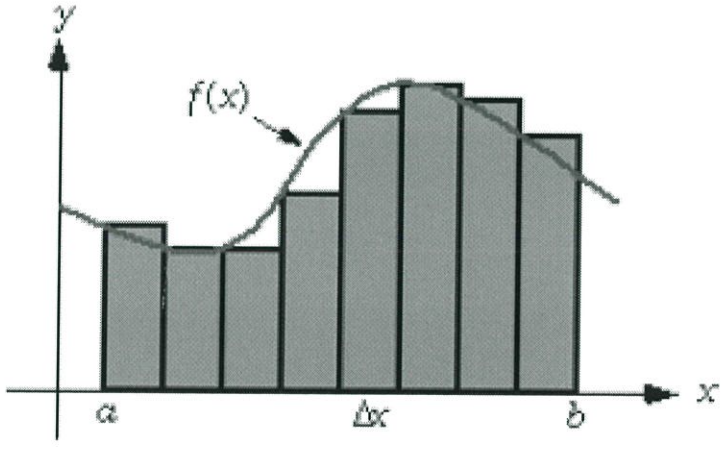
ดังนั้น $$\vec{a} = \frac{d(d\vec{x})}{dt^2} \quad (2.6)$$

$$\vec{V} = \int \vec{a}\, dt \quad (2.7)$$

$$\vec{X} = \int \vec{v}\, dt \quad (2.8)$$

ดังนั้น $$\vec{X} = \int (\int \vec{a}\, dt) dt \quad (2.9)$$

ในการจะที่จะหา $\vec{X}$ ได้นั่น เกิดจากการ อินทิเกรต $\vec{a}$ การหาปริพันธ์เชิงตัวเลขโดยวิธี Trapezoidal



รูปที่ 2.23 ตัวอย่างการอินทิเกรตสัญญาณ [6]

$$\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\Delta x$$

$$\Delta x = \frac{b-a}{n}$$

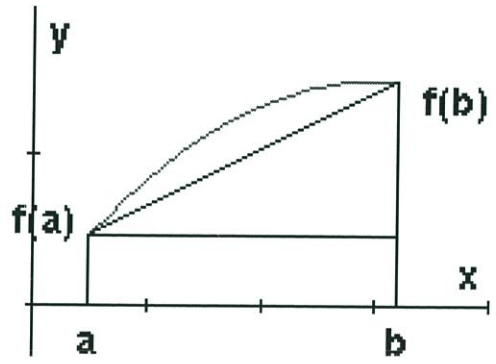
โดยที่ a คือ จุดเริ่มต้นของข้อมูลแกน X

b คือ จุดสุดท้ายของข้อมูลแกน X

n คือ จำนวนเซ็กเม้น

ดังนั้น สามารถหาพื้นที่ใต้กราฟทั้งหมดได้จาก

$$\int_a^b f(x)dx = (b-a)f(a) + \frac{1}{2}(b-a)[f(b) - f(a)]$$



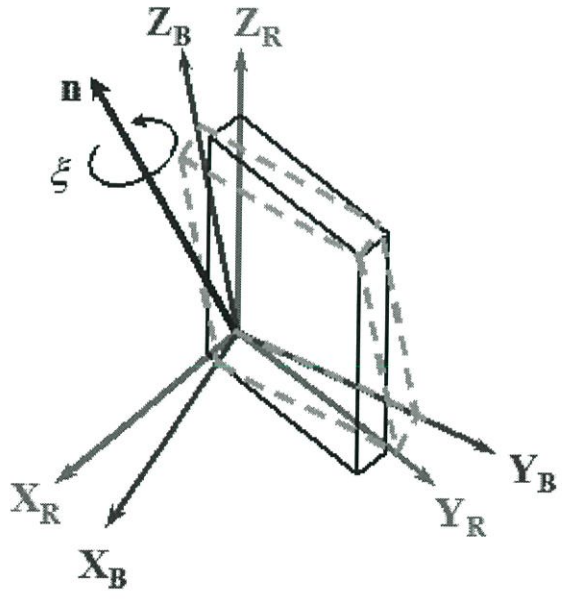
รูปที่ 2.24 ตัวอย่างการอินทิเกรตสัญญาณ [6]

2.9 ทฤษฎีการหมุนของวัตถุ

การหมุนของวัตถุนั้น มีทฤษฎีที่เกี่ยวข้องกับการหมุน 2 ทฤษฎี คือ ควอเทอร์เนียน และมุมของออยเลอร์ ซึ่งใช้ทั้งสองทฤษฎีในการจำลองการหมุนของวัตถุ โดยใช้ ควอเทอร์เนียน ในการคำนวณการหมุน และใช้มุมของออยเลอร์ ในการแสดงทิศทางของวัตถุ

2.9.1 ทฤษฎีควอเทอร์เนียน

ในเชิงกลศาสตร์ การหมุนของวัตถุหนึ่งๆ ที่เกิดขึ้นนั้น จะมีแกนการหมุนเพียงหนึ่งแกน (n) และมุมการหมุนเพียงหนึ่งมุม (ξ) เท่านั้น ดังแสดงในรูปที่ 2.25 โดยจะเห็นได้ว่าวัตถุ (กล่องสี่เหลี่ยม) สามารถหมุนหรือแปลงจากระบบพิกัดหนึ่งไปยังอีกระบบพิกัดหนึ่งได้ โดยการหมุนดังกล่าวอาศัยแกนการหมุน n และมุมการหมุน ξ ซึ่งการวางตัวของวัตถุดังกล่าวสามารถอธิบายได้ด้วยพารามิเตอร์ที่เรียกว่า "ควอเทอร์เนียน" (Quaternion) ที่ประกอบด้วยสมาชิก 4 ตัว การใช้ควอเทอร์เนียนนั้นเป็นที่นิยมมากในเชิงการคำนวณ เนื่องจากไม่มีฟังก์ชันของตรีโกณมิติอยู่เลย ทำให้โอกาสการหารด้วยศูนย์จึงไม่มี อย่างไรก็ตามสมาชิกทั้ง 4 ตัวของควอเทอร์เนียนไม่สื่อความหมายทางกายภาพใดๆ เลย ทำให้ ไม่สามารถตีความได้



รูปที่ 2.25 การหมุนของวัตถุในระบบควอเทอร์เนียน [12]

ควอเทอร์เนียน สามารถนิยามได้ 3 แบบ คือ

1. ควอเทอร์เนียน คือ เวคเตอร์ หนึ่งที่มี 4 องค์ประกอบ

$$\mathring{p} = p_0 + p_1\hat{i} + p_2\hat{j} + p_3\hat{k}$$

2. ควอเทอร์เนียน คือ สเกลาร์ บวกด้วย เวคเตอร์ ที่มี 3 องค์ประกอบ

$$\overset{\circ}{p} = (p_0 + \vec{p}) \text{ โดยที่ } \vec{p} = p_1\hat{i} + p_2\hat{j} + p_3\hat{k}$$

3. ควอเทอร์เนียนคือ complex number ที่มี “imaginary” อยู่สามส่วน

$$\overset{\circ}{p} = p_0 + p_1 i + p_2 j + p_3 k$$

โดยที่

$$i^2 = j^2 = k^2 = ijk = -1$$
$$ij = k \quad jk = i \quad ki = j \quad ji = -k \quad kj = -i \quad ik = -j.$$

ดังนั้น ถ้ามีควอเทอร์เนียน 3 ปริมาณ คือ $\overset{\circ}{p}, \overset{\circ}{q}$ และ $\overset{\circ}{r}$ มารถเขียนได้เป็น

$$\overset{\circ}{\mathrm{p}} = p_0 + ip_1 + jp_2 + kp_3$$
$$\overset{\circ}{\mathrm{q}} = q_0 + iq_1 + jq_2 + kq_3$$
$$\overset{\circ}{\mathrm{r}} = r_0 + ir_1 + jr_2 + kr_3$$

การบวกลบควอเทอร์เนียน

ถ้ามีควอเทอร์เนียน 2 ปริมาณ คือ $\overset{\circ}{p} = p_0 + p_1 i + p_2 j + p_3 k$ ะ $\overset{\circ}{q} = q_0 + q_1 i + q_2 j + q_3 k$ จะได้ว่า

$$\overset{\circ}{p} \pm \overset{\circ}{q} = (p_0 \pm q_0) + (p_1 \pm q_1)i + (p_2 \pm q_2)j + (p_3 \pm q_3)k$$

โดยมีสมบัติการบวก/ลบ คือ

$$\overset{\circ}{p} + \overset{\circ}{q} = \overset{\circ}{q} + \overset{\circ}{p} \qquad \text{Commutative}$$

$$(\overset{\circ}{p} + \overset{\circ}{q}) + \overset{\circ}{r} = \overset{\circ}{p} + (\overset{\circ}{q} + \overset{\circ}{r}) \qquad \text{Associative}$$

การคูณควอเทอร์เนียน

ถ้ามีควอเทอร์เนียน 2 ปริมาณคือ $\overset{\circ}{p} = p_0 + p_1 i + p_2 j + p_3 k$ ่อ $\overset{\circ}{p} = p_0 + \vec{p}$ เมื่อ

$\vec{p} = p_1 i + p_2 j + p_3 k$ และ $\overset{\circ}{q} = q_0 + q_1 i + q_2 j + q_3 k$ หรือ $\overset{\circ}{q} = q_0 + \vec{q}$ เมื่อ $\vec{q} = q_1 i + q_2 j + q_3 k$ จะได้ว่า

$$\begin{aligned}
\overset{\circ}{p}\overset{\circ}{q} &= (p_0q_0 + p_0q_1i + p_0q_2j + p_0q_3k) + (p_1q_0i + p_1q_1ii + p_1q_2ij + p_1q_3ik) + \\
&\quad (p_2q_0j + p_2q_1ji + p_2q_2jj + p_2q_3jk) + (p_3q_0k + p_3q_1ki + p_3q_2kj + p_3q_3kk) \\
&= p_0q_0 + p_0q_1i + p_0q_2j + p_0q_3k + p_1q_0i - p_1q_1 + p_1q_2k - p_1q_3j + \\
&\quad p_2q_0j - p_2q_1k - p_2q_2 + p_2q_3i + p_3q_0k + p_3q_1j - p_3q_2i - p_3q_3 \\
&= (p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3) + (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)i + \\
&\quad (p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1)j + (p_0q_3 + p_1q_2 + p_3q_0 - p_2q_1)k \\
&= p_0q_0 - (p_1q_1 + p_2q_2 + p_3q_3) + p_0(q_1i + q_2j + q_3k) + q_0(p_1i + p_2j + p_3k) + \\
&\quad (p_2q_3 - p_3q_2)i - (p_3q_1 - p_1q_3)j + (p_1q_2 - p_2q_1)k \\
&= [p_0q_0 - \vec{p}\cdot\vec{q}] + [p_0\vec{q} + q_0\vec{p} + \vec{p}\times\vec{q}]
\end{aligned}$$

โดยที่เป็น $p_0q_0 - \vec{p}\cdot\vec{q}$ scalar และ $p_0\vec{q} + q_0\vec{p} + \vec{p}\times\vec{q}$ vector

โดยมีสมบัติการคูณ คือ

$(\overset{\circ}{p}\overset{\circ}{q})\overset{\circ}{r} = \overset{\circ}{p}(\overset{\circ}{q}\overset{\circ}{r})$ Associative

แต่ $\overset{\circ}{p}\overset{\circ}{q} \neq \overset{\circ}{q}\overset{\circ}{p}$ Non - commutative

ควอเทอร์เนียนบริสุทธิ์

คือ ควอเทอร์เนียน ที่ปริมาณสเกลาร์เป็นศูนย์

$$\overset{\circ}{p} = 0 + p_1i + p_2j + p_3k$$

ดังนั้น ถ้าพิจารณาการคูณควอเทอร์เนียน จะพบว่า

$$\overset{\circ}{p}\overset{\circ}{q} = [0 - \vec{p}\cdot\vec{q}] + [0 + 0 + \vec{p}\times\vec{q}] = -\vec{p}\cdot\vec{q} + \vec{p}\times\vec{q}$$

และจะได้ $\overset{\circ}{p}\overset{\circ}{p} = -\vec{p}\cdot\vec{p} + \vec{p}\times\vec{p} = -\vec{p}\cdot\vec{p} = -|\vec{p}|^2$

ขนาดของควอเทอร์เนียน (Magnitude/Norm/Modulus)

ถ้าควอเทอร์เนีย คือ $\overset{\circ}{p} = p_0 + p_1i + p_2j + p_3k$ ดังนั้น Magnitude/Norm/Modulus ของควอเทอร์เนียน จะมีค่าเป็น

$$\left|\overset{\circ}{p}\right| = \sqrt{p_0^{\,2} + p_1^{\,2} + p_2^{\,2} + p_3^{\,2}}$$

ควอเทอร์เนียนหนึ่งหน่วย

คือ ควอเทอร์เนียนที่มีขนาดหนึ่งหน่วย

$$\hat{p} = 1; \qquad \left|\overset{\circ}{p}\right| = \sqrt{p_0^{\,2} + p_1^{\,2} + p_2^{\,2} + p_3^{\,2}} = 1$$

สามารถทำการ normalise ปริมาณควอเทอร์เนียนใดๆ ให้มีปริมาณหนึ่งหน่วยได้โดย

$$\hat{q} = \frac{\overset{\circ}{q}}{\left|\overset{\circ}{q}\right|}$$

ควอเทอร์เนียนคอนจูเกต

ถ้าควอเทอร์เนียน คือ $\overset{\circ}{p} = p_0 + p_1 i + p_2 j + p_3 k$ ดังนั้น ควอเทอร์เนียนคอนจูเกต คือ

$$\overset{\circ}{p}^{*} = p_0 - p_1 i - p_2 j - p_3 k = p_0 - (p_1 i + p_2 j + p_3 k)$$

พิจารณาผลคูณระหว่าง $\overset{\circ}{p}$ และ $\overset{\circ}{p}^{*}$

$$\begin{aligned}
\overset{\circ}{p}\overset{\circ}{p}^{*} &= p_0^{\,2} + (p_1 i + p_2 j + p_3 k)\cdot(-p_1 i - p_2 j - p_3 k) + p_0(p_1 i + p_2 j + p_3 k) + p_0(-(p_1 i + p_2 j + p_3 k)) + \\
&\quad (p_1 i + p_2 j + p_3 k)\times(-(p_1 i + p_2 j + p_3 k)) \\
&= p_0^{\,2} + (p_1 i + p_2 j + p_3 k)\cdot(-p_1 i - p_2 j - p_3 k) \\
&= p_0^{\,2} + p_1^2 + p_2^2 + p_3^2 \\
&= \left|\overset{\circ}{p}\right|^2
\end{aligned}$$

หรือ $\left|\overset{\circ}{p}\right| = \sqrt{\overset{\circ}{p}\overset{\circ}{p}^{*}}$ และคุณสมบัติอื่นๆที่น่าสนใจ ได้แก่ $\overset{\circ}{p}\overset{\circ}{p}^{*} = \overset{\circ}{p}^{*}\overset{\circ}{p}$ และ $(\overset{\circ}{p}\overset{\circ}{q})^{*} = \overset{\circ}{q}^{*}\overset{\circ}{p}^{*}$

อินเวอร์สควอเทอร์เนียน

ถ้าควอเทอร์เนียน คือ $\overset{\circ}{p} = p_0 + p_1 i + p_2 j + p_3 k$ ดังนั้นอนเวอร์สควอเทอร์เนียนคือ $\overset{\circ}{p}^{-1}$
ซึ่งสอดคล้องกับคุณสมบัติ $\overset{\circ}{p}\overset{\circ}{p}^{-1} = \overset{\circ}{p}^{-1}\overset{\circ}{p} = 1$ คูณด้วย $\overset{\circ}{p}^{*}$ จะได้

$$\overset{\circ}{p}^{*}\overset{\circ}{p}\overset{\circ}{p}^{-1} = \overset{\circ}{p}^{*}\overset{\circ}{p}^{-1}\overset{\circ}{p} = \overset{\circ}{p}^{*}$$

นั่นคือ $\overset{\circ}{p}^{-1} = \dfrac{\overset{\circ}{p}^{*}}{\overset{\circ}{p}^{*}\overset{\circ}{p}} = \dfrac{\overset{\circ}{p}^{*}}{\left|\overset{\circ}{p}\right|^2}$ ในกรณีควอเทอร์เนียนหนึ่งหน่วย ซึ่งมี $\left|\overset{\circ}{p}\right|^2 = 1$

ดังนั้น $$\overset{\circ}{p}^{-1} = \overset{\circ}{p}^{*}$$

2.9.2 ทฤษฎีมุมของออยเลอร์

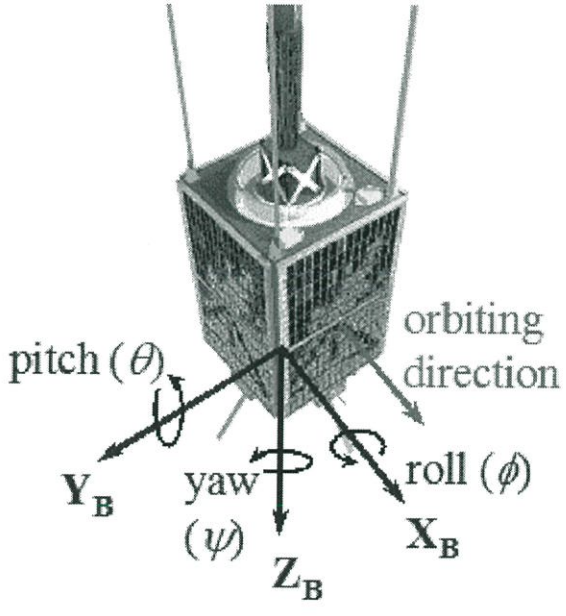
จากการที่ ควอเทอร์เนียน ไม่สามารถจินตนาการหรือสร้างภาพการหมุนได้ จึงมีการพิจารณาให้การหมุนนั้น (หนึ่งแกนและหนึ่งมุม) สามารถอธิบายได้ด้วยการหมุนแบบลำดับ (Sequence rotation) โดยเป็นการหมุนทีละแกนในระบบพิกัดสามมิติ จนครบสามแกน ทั้งนี้จะหมุนแกนใดก่อนหลังก็ได้ ท้ายที่สุดแล้วจะได้ผลลัพธ์เหมือนกัน

ซึ่งมีการใช้วิธีการหมุนของออยเลอร์ (Euler angles rotation) มาใช้ ซึ่งได้แก่มุมโรล (Roll) มุมพิตช์ (Pitch) และมุมยอ (Yaw)

มุมโรลถูกกำหนดให้เป็นมุมของการหมุนรอบแกน X_B

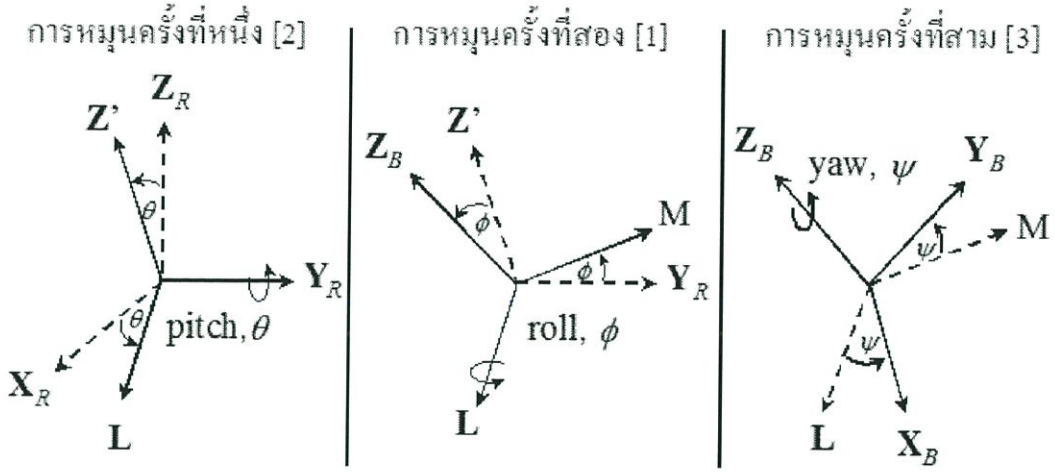
มุมพิตช์ถูกกำหนดให้เป็นมุมของการหมุนรอบแกน Y_B

มุมยอถูกกำหนดให้เป็นมุมของการหมุนรอบแกน Z_B



รูปที่ 2.26 กายภาพของมุมโรล มุมพิตช์ และมุมยอ [12]

จากทฤษฎีการหมุนของออยเลอร์ จะหาการหมุนแบบลำดับ โดยสมมุติให้ต้องการหมุนแบบ 2-1-3 ซึ่งมีความหมายว่าอยากจะหมุนวัตถุรอบแกนที่ 2 (Y_R) ก่อน แล้วตามด้วยการหมุนรอบแกนที่ 1 (X_R) และการหมุนรอบแกนที่ 3 (Z_R) เป็นลำดับ



รูปที่ 2.27 กายภาพของการหมุนแบบลำดับ 2-1-3 [12]

จากรูปที่ 2.27 ในการหมุนครั้งที่หนึ่ง (รอบแกน Y_R) ด้วยขนาดมุมหมุนพิตช์ (θ) จะสังเกตว่าแกน Y_R ยังชี้ไปในทิศทางเดิม แต่แกน X_Rและ Z_R ชี้ไปในทิศทางใหม่เป็น L และ Z^1 ตามลำดับ

การหมุนครั้งที่สอง (รอบแกน L ซึ่งเป็นแกนใหม่ของ X_R) ด้วยขนาดมุมหมุนโรล($\emptyset$) จะสังเกตว่าแกน L ยังชี้ไปในทิศทางเดิม แต่แกน Y_R และ Z^1 ชี้ไปในทิศทางใหม่เป็น M และ Z_B) ตามลำดับ

การหมุนครั้งสุดท้าย (รอบแกน Z_Bซึ่งเป็นแกนใหม่ของ Z_R) ด้วยขนาดมุมหมุนยอ (ψ) จะสังเกตว่าแกน Z_B) ยังชี้ไปในทิศทางเดิม แต่แกน L และ M ชี้ไปในทิศทางใหม่เป็น X_B และ Y_B ตามลำดับ

ท้ายที่สุดจะพบว่าเมื่อการหมุนทั้งสามครั้งสิ้นสุดลง ระบบพิกัดอ้างอิงถูกแปลงมาเป็นระบบพิกัดของตัววัตถุ ซึ่งมุมโรล มุมพิตช์ และมุมยอเป็นพารามิเตอร์ที่ใช้อธิบายการทิศทางการวางตัวระหว่างระบบพิกัดทั้งสอง

2.10 โปรแกรมที่เกี่ยวข้อง

2.10.1 Microsoft Visual Studio

Microsoft Visual Studio คือ Integrated Development Environment พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอปพลิเคชัน และเว็บเซอร์วิส ระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์ วินโดวส์ พ็อคเกตพีซี สมาร์ทโฟน และ เว็บเบราว์เซอร์ ในปัจจุบัน Visual Studio นั้นสามารถใช้ภาษาโปรแกรมที่เป็นภาษาดอตเน็ต ในโปรแกรมเดียวกัน เช่น VB.NET C++ C# J# เป็นต้น

วิชวลสตูดิโอ 2010 ซึ่งเป็นรุ่นหลังจาก 2011 ได้แบ่งเป็นรุ่นดังต่อไปนี้

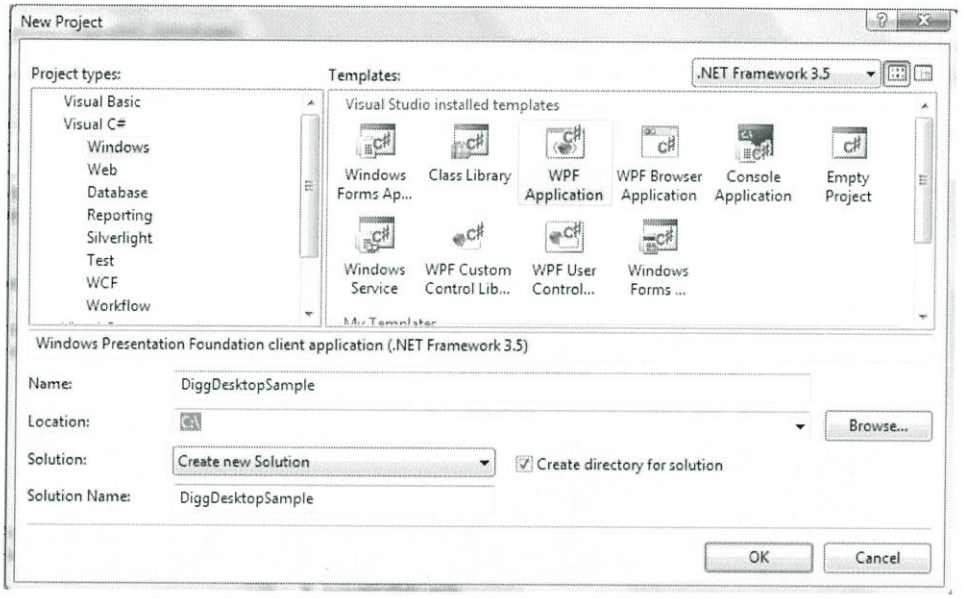
- Visual Studio Standard Edition
- Visual Studio Premium Edition
- Visual Studio Professional Edition
- Visual Studio Team Foundation Server: Trial
- Visual Studio Express Edition
- Visual Studio SDK
- Visual Studio .NET Framework 4.0
- Visual Studio Test Professional Edition
- Visual Studio Team Explorer - ISO
- Visual Studio Team Explorer Everywhere
- Visual Studio 2Lab Management
- Visual Studio Remote Debugger
- Visual Studio F#
- Visual Studio Visualization & Modeling SDK
- Visual Studio Agents
- Visual Studio Tools for Office Runtime
- Visual Studio Shell (Integrated)
- Visual Studio Shell (Isolated)

Visual Studio 2011 Beta ซึ่งเป็นหลังจาก 2012 RC ได้แบ่งเป็นรุ่นดังต่อไปนี้

- Visual Studio Ultimate
- Visual Studio Premium
- Visual Studio Professional
- Visual Studio Test Professional

Visual Studio 2012 RC ซึ่งเป็นรุ่นล่าสุดได้แบ่งเป็นรุ่นดังต่อไปนี้

- Visual Studio Ultimate
- Visual Studio Premium
- Visual Studio Professional
- Visual Studio Test Professional
- Visual Studio Team Foundation Server



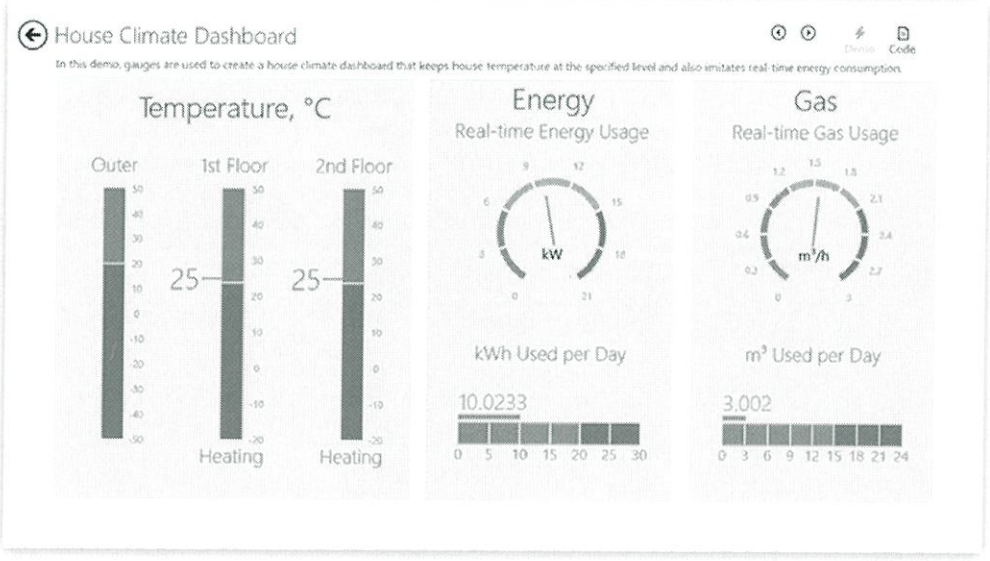
รูปที่ 2.28 ขั้นตอนการสร้างโปรเจค ของ Windows Form Application

.NET Windows Form Application เขียนโปรแกรมบน Windows Form Application ด้วย .NET Framework ใน Application บน Visual Studio นั้น Windows Form ถือเป็น Project พื้นฐานที่สามารถพัฒนาโปรแกรมที่ทำงานบน Windows ได้ง่ายและรวดเร็วที่สุดก็ว่าได้ เพราะเป็นการออกแบบรูปแบบ GUI การใส่ Control หรือกำหนด Event ต่างๆ สามารถสร้างเหตุการณ์ต่างๆ ได้จาก Properties ของ Control และเค้าโครงการเขียนนั้นก็มีพื้นฐานมาจากภาษา Visual Basic 6.0 นักโปรแกรมเมอร์ที่พัฒนาโปรแกรมด้วย VB6 มาก่อนหน้านี้จะสามารถต่อยอดการเขียนได้ง่าย รูปแบบคำสั่งที่เป็นภาษา VB.NET ไม่มีความซับซ้อน ซึ่งใน .NET Framework นี้สามารถพัฒนาโปรแกรมให้มีความสามารถและการทำงานได้หลากหลาย และยังสามารถเขียนเพื่อใช้งานร่วมกับ Applicationอื่นๆที่พัฒนาด้วย.NETFrameworkได้

WPF ย่อมาจาก Windows Presentation Foundation เป็นแนวคิดการพัฒนา UI (User Interface) อีกแนวหนึ่งที่มีความยืดหยุ่นสูง มีกระบวนการ Data Binding ที่ยืดหยุ่นสูง ทำให้สามารถเขียน UI ที่ซับซ้อนได้สะดวก โดยสามารถออกแบบในมุมมองเสมือนว่าเป็น Projection ของข้อมูล ได้ ข้อดีคือ สามารถเขียน View Model Object เพื่อควบคุมการไหลเวียนของข้อมูลครั้งเดียว โดยที่ UI จะเป็นเสหมือนหน้ากากครอบ โดยจะเปลี่ยนไปเป็นรูปแบบใดก็ได้ ตราบเท่าที่

โปรแกรม Bind กับ View Model Object ถูกต้อง ทำให้สามารถที่จะ Unit Test ตัว UI ได้โดยผ่าน View Model Object ที่ออกแบบมาเป็นอย่างดี

WPF ยังสนับสนุน UI ที่สร้างสรรค์ อนุญาตให้ Graphic Designer ออกแบบได้ โดยผ่าน XAML ที่สะดวกที่สุด ในการที่จะทำให้โปรแกรมมีที่สวยงาม มี



รูปที่ 2.29 ตัวอย่างหน้าจอ UI

WPF สนับสนุนการเขียนโปรแกรมบน Design Pattern แบบ MVVM (Model-View-ViewModel)

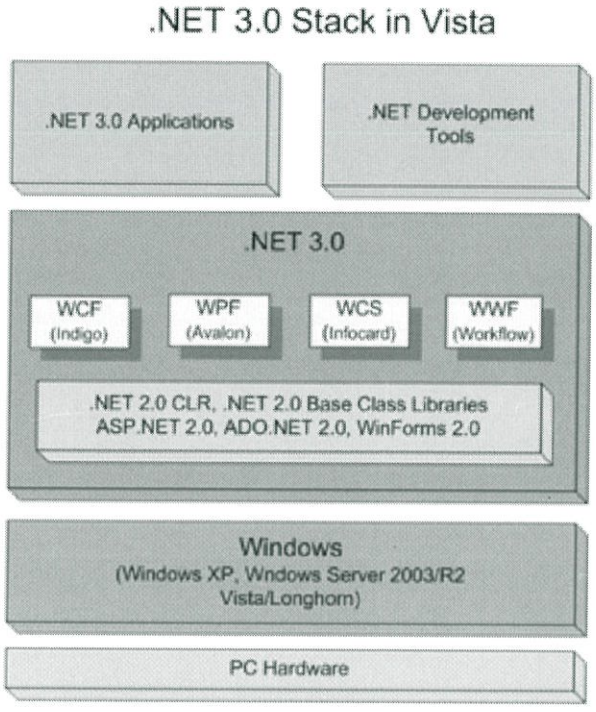
Window1.xaml

```
<Window x:Class="DiggDesktopSample.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Digg="clr-namespace:DiggDesktopSample"
    Title="Digg Desktop Version" Height="600" Width="700">

    <Grid>
        <Digg:Page/>
    </Grid>

</Window>
```

รูปที่ 2.30 ตัวอย่างโค้ดของโปรแกรม WPF

Visual Studio เป็นเครื่องมือในการพัฒนา WPF โดยที่สามารถใช้สำหรับเขียนโค้ด ตั้งแต่ View (XAML) ไปจนถึง ViewModel ไปจนถึงระดับของ Database เพียงแต่ไม่มีเครื่องมือ Visual เพื่ออำนวยความสะดวกในการออกแบบ ในขณะที่ Expression Blend ใช้สำหรับเป็นตัวช่วยสร้าง

XAML เพื่อสร้าง Graphic ของ UI ไปจนถึงการทำ Animation Storyboard และออกแบบ Flow ของ UI



รูปที่ 2.31 โครงสร้างของ WPF ใน .NET Framework [13]

ไมโครซอฟท์มีความพยายามจะนำ WPF มารันบนเบราเซอร์ผ่านตัว Plug-In ที่ชื่อ Silverlight เพื่อให้นักพัฒนาสามารถสร้างงาน WPF บนเบราเซอร์ได้ทำให้สามารถสร้าง Window Application หรือ Web Application ที่รันผ่าน Plug-In Silverlight

ส่วนของการพัฒนา Microsoft เลือกใช้ภาษาที่อยู่บนพื้นฐานของ XML ทำให้นักพัฒนาสามารถปรับแต่งมุมมองของ Source

ข้อดีของ WPF

1. สามารถแบ่งหน้าที่การทำงานระหว่าง Designer และ Programmer ได้อย่างชัดเจน
2. มีกราฟฟิกที่สวยงามสร้าง control ใช้เองได้
3. มี Style (คล้ายๆ CSS ของ Web) ซึ่งสามารถสร้าง Control หรือ Window เดียวแล้วเปลี่ยน Style ก็จะทำให้ Look & Feel เปลี่ยนทั้งหมดโดยที่ไม่ต้องแก้โค้ด

2.10.2 ไลบรารี (Library)

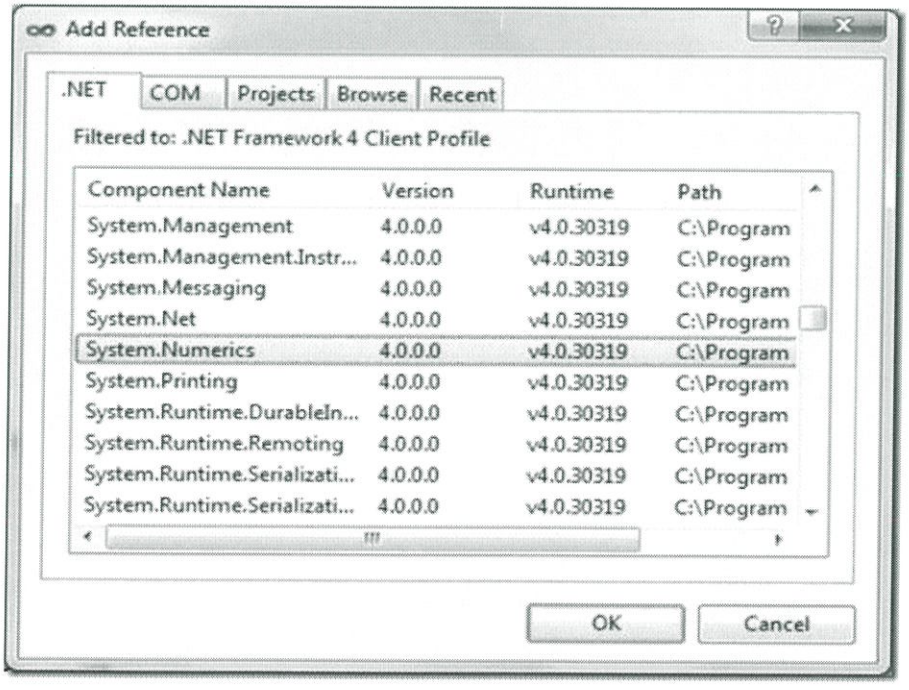
ในคอมพิวเตอร์ ไลบรารี (Library) เป็นชุดของอ๊อบเจคขนาดเล็กที่ได้รับการจัดเก็บสำหรับการใช้ตามโอกาส ส่วนใหญ่บ่อย โปรแกรมใน Source Code หรือรูปแบบ Object Code, ไฟล์ข้อมูล, สคริปต์, เทมเพลท (Template), ฟอนต์ และหน่วยการจัดเก็บทางกายภาค เช่น เทป

1) ไลบรารีโปรแกรมเป็นชุดของ ฟังก์ชันประจำ (Routine) ของโปรแกรมคอมไพล์และใช้ใหม่ได้ที่โปรแกรมเมอร์สามารถ "เรียก" เมื่อเขียนคำสั่ง ดังนั้นโปรแกรมเมอร์ไม่ต้องเขียนขึ้นใหม่ Dynamic Link Library (DLL) เป็นประเภทหนึ่งของไลบรารีโปรแกรม อีกประเภทของโปรแกรมไลบรารีคือ ไลบรารี Class ซึ่งเก็บฟังก์ชันประจำเป็นข้อกำหนด Class ใน Oobject-Oriented Programming (OOP) ส่วนประกอบ Graphical User Interface (GUI) เช่น Scroll Bar, Button และฟังก์ชันประจำได้รับการเก็บในไลบรารี Class

2) ไลบรารี Storage เป็นชุดของตัวกลางจัดเก็บทางกายภาค เช่น เทป หรือดิสก์ และมีการเข้าถึง ตัวอย่างเช่น ไลบารีเทปบรรจุตลับเทปและกลไกที่เคลื่อนย้ายเทปเข้าและออกจากไดร์ฟ ที่มีการอ่านหรือปรับปรุงเนื้อหา

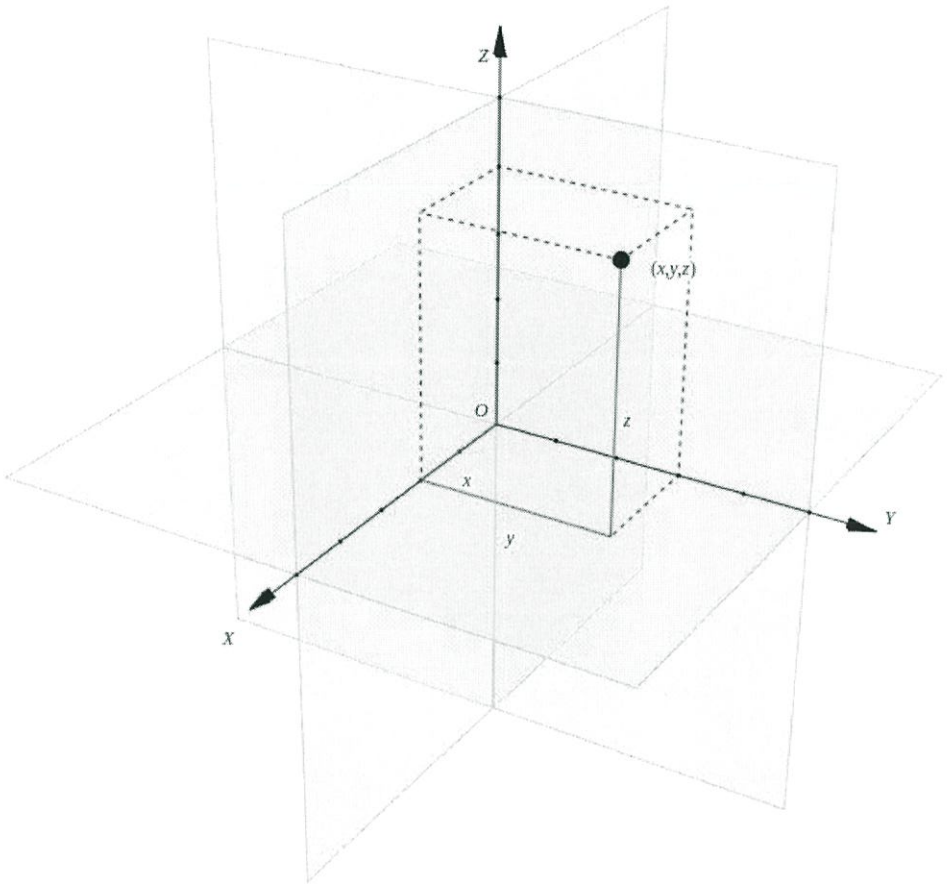
3) ไลบรารีข้อมูลคือ พื้นที่ของ Data Center (ระบบคอมพิวเตอร์พื้นที่ส่วนกลางตัวครอบพื้นที่ส่วนกลางและอุปกรณ์) ที่เก็บตัวกลางจัดเก็บได้รับบรรจุ บางครั้ง ผู้ให้บริการออนไลน์เรียกว่าไดเรคทอรีบนเครื่องแม่ข่ายที่เก็บไฟล์สำหรับดวน์โหลดเป็นไลบรารีข้อมูล

4) ไลบรารีเสมือนเป็นเวอร์ชันออนไลน์อย่างง่ายของไลบรารีตามแบบแผน หนังสือและเอกสารมีให้บนเว็บ และอาจจะอ่านออนไลน์หรือดาวน์โหลด



รูปที่ 2.32 ตัวอย่างการนำเข้าไลบรารี

2.10.3 3D Tools for the Windows Presentation Foundation (WPF)



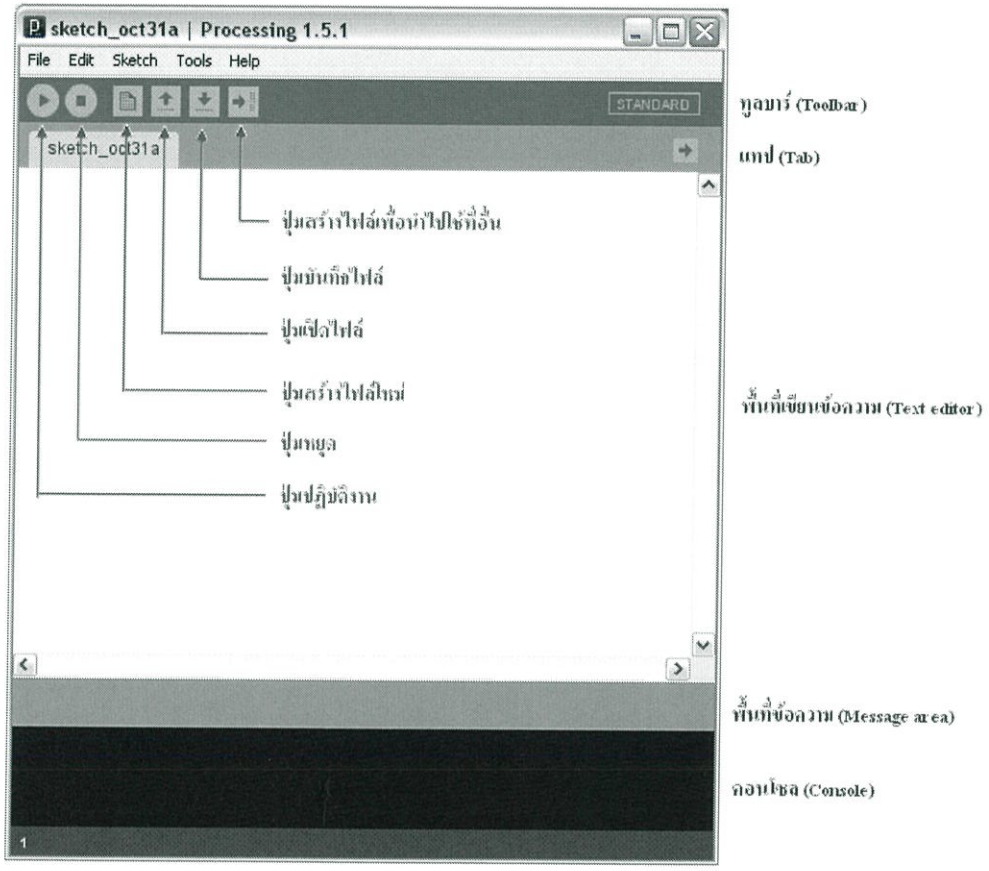
รูปที่ 2.33 ตัวอย่างการจำลองรูปแบบ 3D [13]

3D Tools เป็นไลบรารี คือชุดของคำสั่งที่ช่วยให้นักพัฒนาเรียกใช้คุณสมบัติ สามมิติใน Windows Presentation Foundation ภายในประกอบด้วย

- ชุดคำสั่งที่ใช้ในการติดต่อ 2D และ 3D
- Trackball class ชุดคำสั่งการหมุน กล้อง โดยใช้ เม้าส์
- ScreenSpaceLines3D ใช้ในการวาดเส้นบนพิกัด 3D
- Texture Coordinate Generators

2.10.4 Processing

Processing เป็นซอฟท์แวร์ระบบเปิด เหมาะสำหรับผู้ต้องการพัฒนาโปรแกรมเกี่ยวกับการสร้างภาพเคลื่อนไหวและการมีปฏิสัมพันธ์ ซึ่งมีหน้าตาคล้ายกับโปรแกรม Arduino และมีพื้นฐานการเขียนมาจากภาษา C/C++ นอกจากนี้ยังมีไลบรารีมากกว่า 100 ไลบรารีอีกด้วย



รูปที่ 2.34 ตัวอย่างโปรแกรม Processing [13]

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การออกแบบฮาร์ดแวร์

ในส่วนของเซนเซอร์เซนเซอร์ Accelerometer Gyroscope and Magnetometer นั้น ออกแบบให้มีการเชื่อมต่อกับไมโครคอนโทรลเลอร์ Arduino Nano (ATmega328P) ผ่านทาง I2C โดยใช้พอร์ต Serial Data (SCL) และ Serial Clock (SDA) กับ ตัวส่งสัญญาณผ่านระบบ Wireless Lan

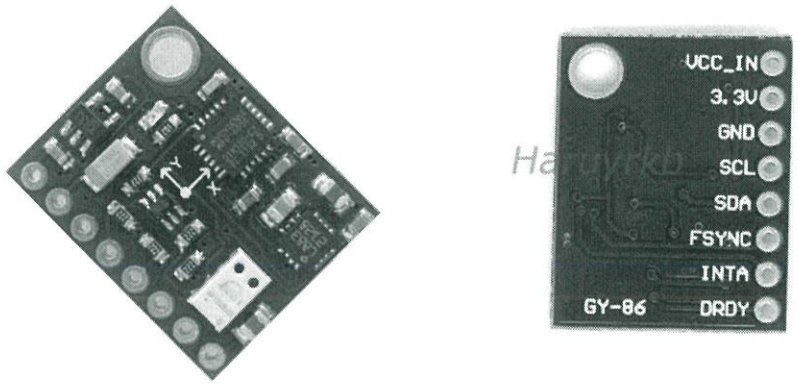
3.1.1 หลักการทำงานโมดูลเซนเซอร์ รุ่น GY-86 (MPU6050 HMC5883L MS5611)

โมดูลนี้เป็นเซนเซอร์วัดค่าความเร่งจากแรงโน้มถ่วงของโลก (MPU6050) วัดความเร็วเชิงมุม (MPU6050) วัดสนามแม่เหล็ก (HMC5883L) และค่าความดัน (MS5611) โดยใช้การสื่อสารผ่าน I2C โดยมีสเกลในการวัดความเร็วเชิงมุม 4 ช่วงดังนี้ ± 250°/s, ± 500°/s, ± 1000°/s และ ± 2000°/s สเกลของความเร่งคือ 4 ช่วงดังนี้ ± 2g, ± 4g, ± 8g และ ± 16g (1g เท่ากับ 9.8 m/s^2) เซนเซอร์สนามแม่เหล็ก จะอ่านค่าความเข้มของสนามแม่เหล็กรอบเซนเซอร์

คุณสมบัติพื้นฐานของโมดูลเซนเซอร์ รุ่น GY-86 (MPU6050 HMC5883L MS5611)

Gyroscope 3 แกน Accelerometer 3 แกน Magnetometer 3 แกน และความดัน

- แรงดันไฟต่ำสุด 3V
- แรงดันไฟสูงสุด 5V
- Gyroscope Range: ± 250°/s, ± 500°/s, ± 1000°/s, ± 2000°/s
- Acceleration Range: ± 2g, ± 4g, ± 8g, ± 16g
- Magnetometer Field Range: ±8 Gauss
- การเชื่อมต่อผ่านโปรโตคอลมาตรฐาน I2C
- อุณหภูมิในการปฏิบัติการที่ -40 องศาเซลเซียส ถึง +85 องศาเซลเซียส
- แรงดันไฟในการปฏิบัติการที่ 3-5V
- มีขนาด 2.2 x 1.7 x 0.3 cm^3



รูปที่ 3.1 โมดูลเซนเซอร์ รุ่น GY-86 (MPU6050 HMC5883L MS5611) [7]

ตารางที่ 3.1 รายละเอียดพอร์ตเชื่อมต่อของโมดูลเซนเซอร์รุ่น GY-521

ชื่อขา	หน้าที่
VCC_IN	ขาต่อไฟเลี้ยงกระแสตรง 3 – 5 V (โวลต์)
3.3V	ขาต่อไฟเลี้ยงกระแสตรง 3.3 V
GND	กราวด์
SCL	ขาการกำหนดรับส่งข้อมูล(Serial clock)
SDA	ขารับส่งข้อมูล Serial Data (SDA)
FSYNC	เชื่อมต่อสัญญาณซิงค์โคไนท์จากภายนอก
INTA	อินเทอรัพ
DRDY	รับค่าอนาล็อก

3.1.2 หลักการทำงานของโมดูล Wireless Lan ที่ใช้งาน

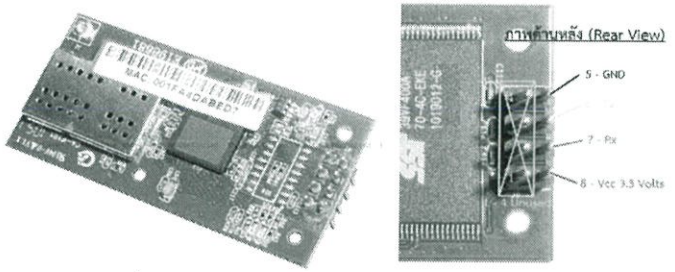
3.1.2.1 โมดูล Wireless Lan รุ่น TNS-WII3

M03 (TNS-WII3 หรือ TNSWII3) - LVTTL UART to WiFi คือ โมดูลที่ใช้สำหรับแปลงการรับส่งข้อมูลในรูปแบบ UART เป็นการรับส่งข้อมูลในรูปแบบของ Wireless LAN หรือ WiFi (IEEE 802.11b/g)

คุณสมบัติพื้นฐานของ Wireless Lan รุ่น TNS-WII3

- Interface
  - 2x4 ขาของการเชื่อมต่อ
  - สามารถรับส่งข้อมูลผ่านทาง UART ที่ความเร็ว 1200 – 115200 bps
  - รองรับ Hardware Flow Control RTS/CTS
  - ใช้ไฟฟ้ากระแสตรง แรงดัน 3.3V
- Wireless
  - รองรับการรับส่งข้อมูลผ่าน Wireless ตามมาตรฐาน IEEE 802.11b/g
  - ใช้ช่วงความถี่ 2.412 – 2.484 GHz
  - รองรับการใช้งานแบบ AdHoc และ Infrastructure

- รองรับมาตรฐานความปลอดภัย WEP64/WEP128/TKIP/CCMP(AES)/ WEP/WPA-PSK/WPA2-PSK
- รองรับ Network Protocol แบบ TCP/UDP/ICMP/DHCP/DNS/HTTP



รูปที่ 3.2 โมดูล Wireless Lan รุ่น TNS-WII3

ตารางที่ 3.2 รายละเอียดพอร์ตเชื่อมต่อของโมดูล Wireless Lan รุ่น TNS-WII3

ชื่อขา	หน้าที่
VCC	ขาต่อไฟเลี้ยงกระแสตรง 3.3 V
GND	กราวด์
RX	ขารับข้อมูล
TX	ขาส่งข้อมูล

3.1.2.2 โมดูล Wireless Lan รุ่น RN-XV WiFly Module (RN-171)

RN-XV WiFly Module หรือเรียกสั้นๆ ว่า WiFly เป็นโมดูลสำหรับเชื่อมต่อสัญญาณ WiFi มาตรฐาน 802.15.4 และ 802.11b/g ประมวลผลด้วยระบบ 32 บิต ช่วยให้จัดการกับการเชื่อมต่อผ่านระบบ WiFi ได้ง่าย โดยตัวโมดูลมีลักษณะคล้ายกับ XBee โมดูลสื่อสารข้อมูลอนุกรมไร้สายผ่านคลื่นวิทยุ 2.4GHz สามารถใช้ซ็อกเก็ตหรือแผงวงจรเชื่อมต่อร่วมกันได้ WiFly ใช้การเชื่อมต่อแบบ UART ติดต่อบอร์ดไมโครคอนโทรลเลอร์

คุณสมบัติพื้นฐานของ Wireless Lan รุ่น RN-XV WiFly Module (RN-171)

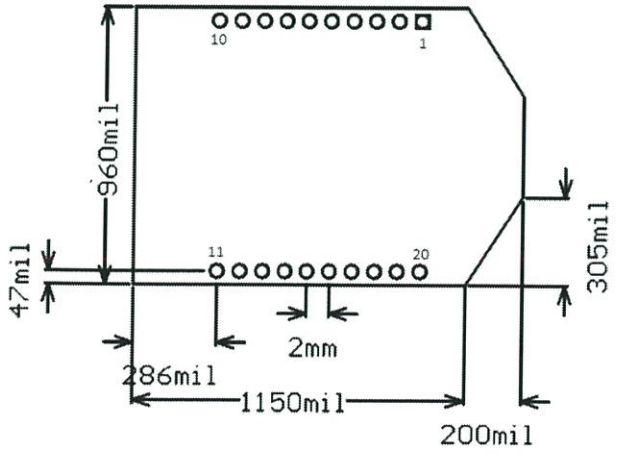
- ขาการเชื่อมต่อตามมาตรฐาน 802.15.4 XBee footprint
- รองรับ TCP/IP Stack Includes DHCP, UDP, DNS, ARP, ICMP, HTTP Client, FTP Client และ TCP
- ปรับกำลังในการส่งได้ตั้งแต่ 0-12 dBm
- การเชื่อมต่อแบบอนุกรมยูอาร์ท (TTL UART)
- ความเร็วข้อมูล(Host Data Rate) 464 Kbps บน UART
- รองรับโหมด AdHoc และ Infrastructure
- 8 General Purpose Digital I/O
- Real-Time Clock for Time-Stamping, Auto-Sleep, and Auto-Wakeup Modes
- รับไฟ 3.3 V จากแหล่งจ่ายไฟ (Power Supply)
- มีเสาอากาศในตัว

ตารางที่ 3.3 คุณสมบัติเฉพาะตัวทางไฟฟ้าของโมดูล Wireless Lan รุ่น RN-XV WiFly Module (RN-171)

Supply Voltage	Min	Typ.	Max.	Unit
Input Power	3.0	3.3	3.7	V
Power Consumption				
Sleep		4		uA
Standby (Doze)		15		mA
Connected (Idle, RX)		40		mA
Connected (TX)		180 at 10dBm		mA

ตารางที่ 3.4 Radio Characteristics

Parameter	Specifications
Frequency	2402 ~ 2480MHz
Modulation	802.11b Compatibility : DSSS(CCK-11, CCK-5.5, DQPSK-2, DBPSK-1) 802.11g : OFDM (Default)
Channel intelvals	5MHz
Channels	1 – 14
Transmission rate (Over the air)	1-11Mbps for 802.11b / 6 – 54Mbps for 802.11g
Receive sensitivity	-83dBm Typ.
Output level (Class1)	0dBm to +12dBm (Software Configurable)



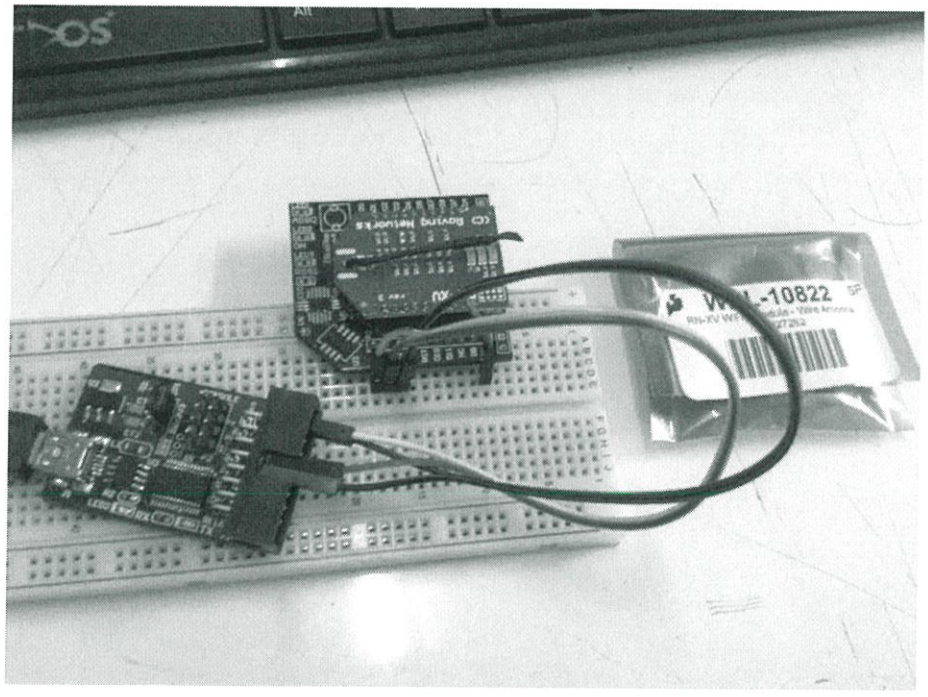
รูปที่ 3.3 ขนาดทางกายภาพ (1 mil = 0.0254 millimeters)

ตารางที่ 3.5 คำอธิบายขาของ Wireless Lan รุ่น RN-XV WiFly Module (RN-171)

Pad Number	Signal Name	Description	Direction
1	VDD_3V3	3.3V Regulated Power Input to The Module	POWER
2	UART_TX	UART TX, 8mA Drive, 3.3V Tolerant	OUT
3	UART_RX	UART RX, 3.3V Tolerant	IN
4	GPIO 8	GPIO, 24mA drive, 3.3V Tolerant	IN / OUT
5	RESET	Optional Module Reset Signal (Active Low), 100k Pull Up, Apply Pulse of at Least 160us, 3.3V Tolerant	INPUT
6	GPIO 5	GPIO, 24mA Drive, 3.3V Tolerant	OUT
7	GPIO 7	GPIO, 24mA Drive, 3.3V Tolerant	IN / OUT
8	GPIO 9	Enable AdHoc Mode, Restore Factory Defaults, 8mA Drive, 3.3V Tolerant	IN / OUT
9	GPIO 1	GPIO, 8mA Drive, 3.3V Tolerant	IN / OUT
10	GND	Ground	GND
11	GPIO 14	GPIO, 8mA Drive, 3.3V Tolerant	IN / OUT
12	UART_RTS	UART RTS Flow Control, 8mA Drive, 3.3V Tolerant	OUT
13	GPIO 4/SEN 6	GPIO, 24mA Drive, 3.3V Tolerant/ADC Input , (3.3V Tolerant) Defaults to GPIO 4	IN / OUT
14	Not Used		No Connect
15	GPIO 6/SEN 7	GPIO, 24mA Drive, 3.3V Tolerant/ADC Input, (3.3V Tolerant). Defaults to GPIO 6	POWER
16	UART_CTS	UART CTS Flow Control, 3.3V Tolerant	IN

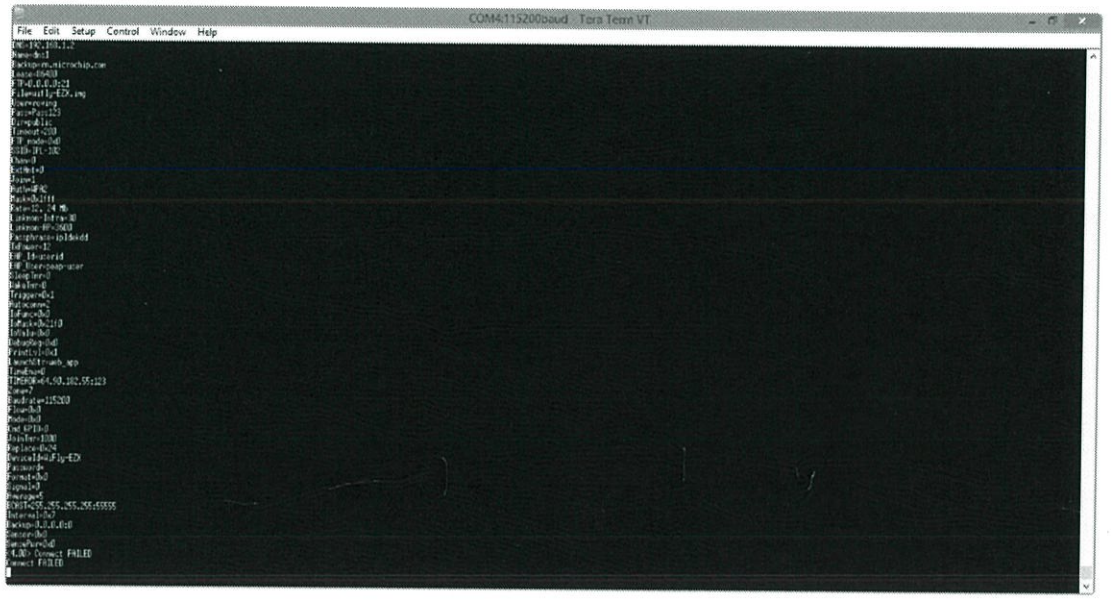
17	SENSOR 5	Sensor Interface, Analog Input to Module, (3.3V Tolerant)	INPUT
18	GPIO 3/SEN 4	GPIO, 8mA Drive, 3.3V Tolerant/ADC Input (3.3V Tolerant). Defaults to GPIO 3	IN / OUT
19	GPIO 2/SEN 3	GPIO, 8mA Drive, 3.3V Tolerant/ADC Input (3.3V Tolerant). Defaults to SEN 3	IN / OUT
20	SEN 2	Sensor Interface, Analog Input to Module, 3.3V Tolerant	INPUT

การตั้งค่า Wireless Lan รุ่น RN-XV WiFly Module (RN-171)



รูปที่ 3.4 การต่อ WiFly กับ Serial UART เพื่อตั้งค่า

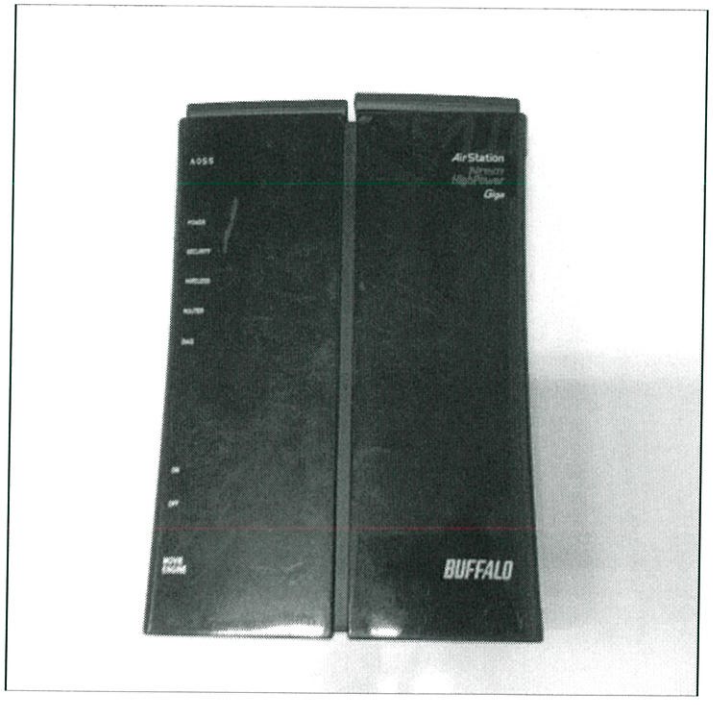
การตั้งค่านั้นสามารถตั้งค่า ตัว WiFly ด้วยโปรแกรม Tera Term Version 4.80 โดยผ่านทาง UART



รูปที่ 3.5 การตั้งค่าโมดูล Wireless Lan ผ่านโปรแกรม Tera Term Version 4.80

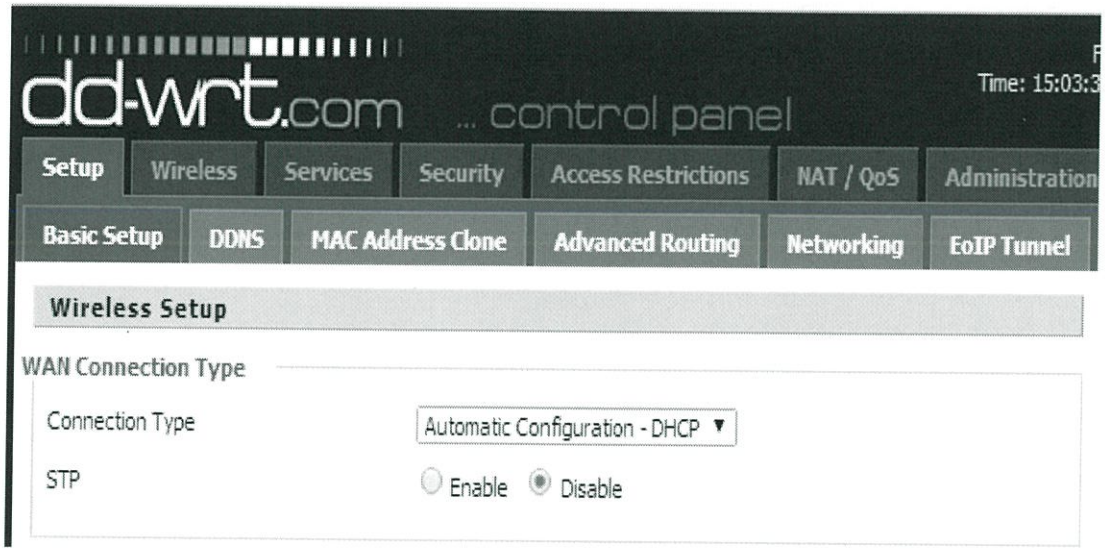
การตั้งค่า Router เพื่อเชื่อมต่อกับ WiFly

Router รุ่น Buffalo WZR-HP-G300NH โดยใช้ Firmware ของ dd-wrt v24-sp2 โดยมีขั้นตอนดังนี้



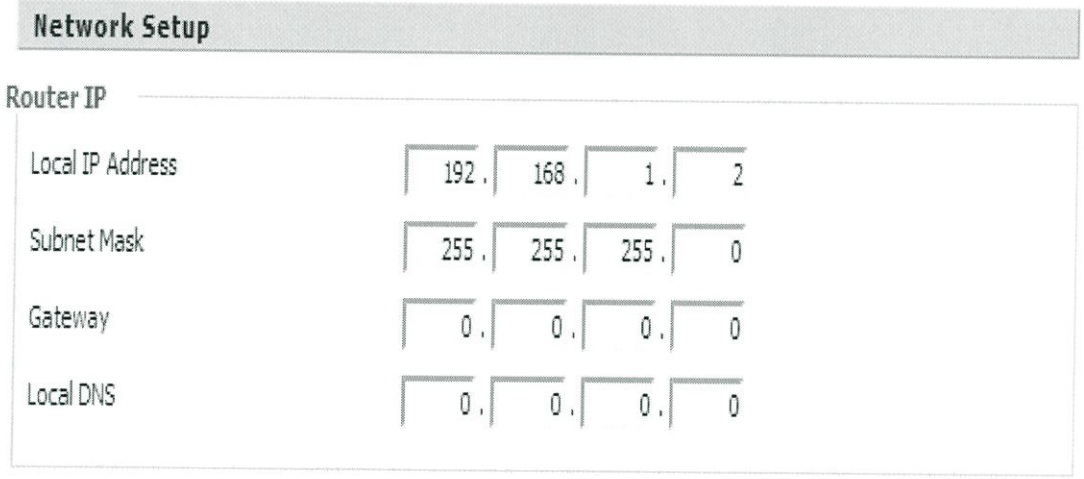
รูปที่ 3.6 Router รุ่น Buffalo WZR-HP-G300NH

1.ตั้งค่า WAN เป็นแบบ Auto ดังแสดงในรูปที่ 3.7



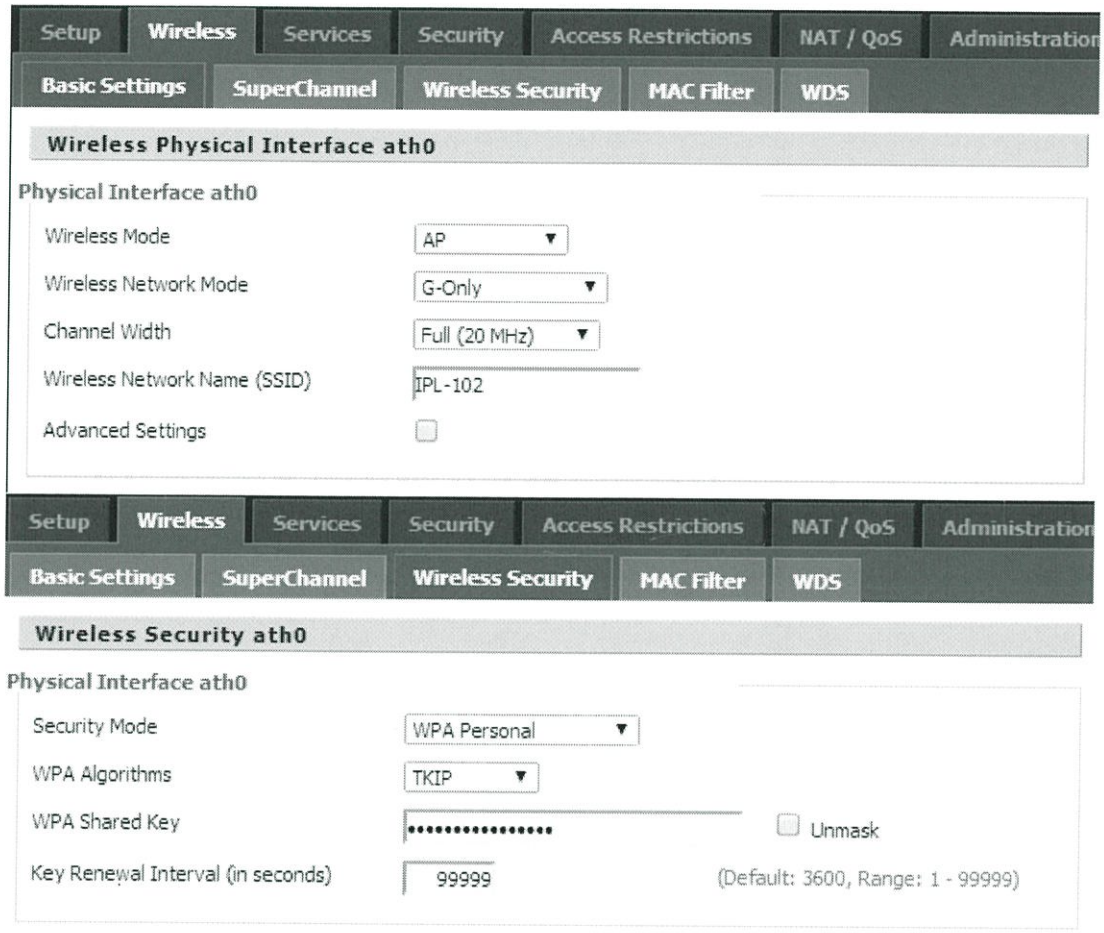
รูปที่ 3.7 การตั้งค่าเร้าเตอร์เพื่อเชื่อมต่อกับ WiFly

2.กำหนด IP Address Router เป็น 192.168.1.2/24



รูปที่ 3.8 การตั้งค่าเร้าเตอร์เพื่อเชื่อมต่อกับ WiFly

3.ตั้งค่าสัญญาณ Wireless ของ Router ดังรูป



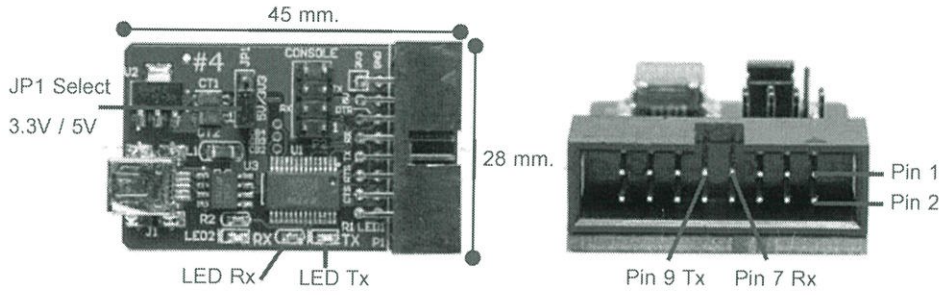
รูปที่ 3.9 การตั้งค่า Router เพื่อเชื่อมต่อกับ WiFly

3.1.3 หลักการทำงานของ Serial UART to USB Mini B Converter V4.0

คุณสมบัติพื้นฐานของ Serial UART to USB Mini B Converter V4.0

- เปลี่ยนสัญญาณจาก UART (TTL) เป็น USB (Connector เป็นแบบ USB Mini B)
- เชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับ USB Bus โดยตรง ไม่จำเป็นต้องแปลงแรงดันเป็น RS-232
- สามารถใช้ในการ Upgrade Firmware ผ่านพอร์ตอนุกรม โดยไม่ต้องมีแหล่งจ่ายไฟภายนอก
- ใช้ต่อเข้ากับอุปกรณ์ที่สื่อสารแบบ TTL เข้ากับคอมพิวเตอร์ ผ่านทาง USB เช่น GPS,RFID Reader , Barcode Reader , Magnetic Reader , ขา Tx/Rx UART ของไมโครคอนโทรเลอร์
- สามารเลือกระดับแรงดันของสัญญาณอนุกรม ให้เป็น 3.3 V หรือ 5 V ได้โดยการเสียบ Jumper ที่ JP1

- มีวงจร Regulator แปลงแรงดัน 5 V จาก USB ให้เป็น 3.3 V จ่ายกระแสได้สูงสุด 1 แอมแปร์ (กระแสที่จ่ายได้จะถูกจำกัดจากพอร์ต USB ซึ่งขึ้นอยู่กับคอมพิวเตอร์ หรือคอมพิวเตอร์โน๊ตบุ๊ค) สามารถนำไปใช้ต่อกับบอร์ดภายนอกได้
- สามารถต่อแรงดัน 5 V จากพอร์ต USB ไปใช้กับบอร์ดภายนอกได้
- PIN Function TX / RX / CTS / RTS / DTR / 3.3 V / 5 V / GND
- Driver รองรับการทำงาน Windows 98SE/ME/2000/XP/7



รูปที่ 3.10 ตำแหน่งขาที่ใช้งานของ Serial UART to USB Mini B Converter V4.0

ตารางที่ 3.6 รายละเอียดขาที่ใช้งานของ Serial UART to USB Mini B

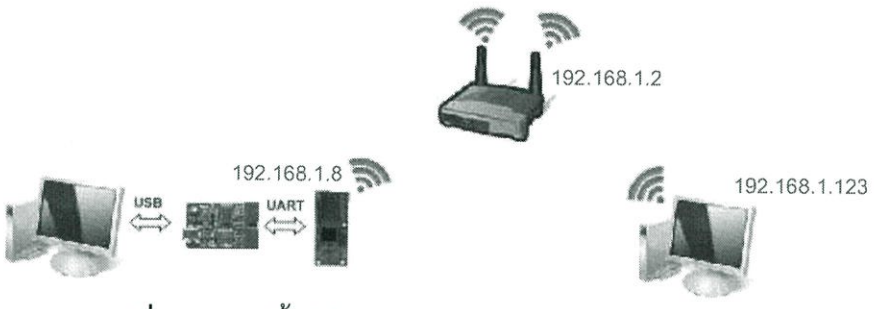
ขาที่	ชื่อขา	หน้าที่
1	3v3	แรงดัน 3.3 V
2	GND	กราวด์
7	Rx	รับสัญญาณ
9	Tx	ส่งสัญญาณ

3.1.4 ตั้งค่าการเชื่อมโมดูล Wireless Lan กับ Access Point

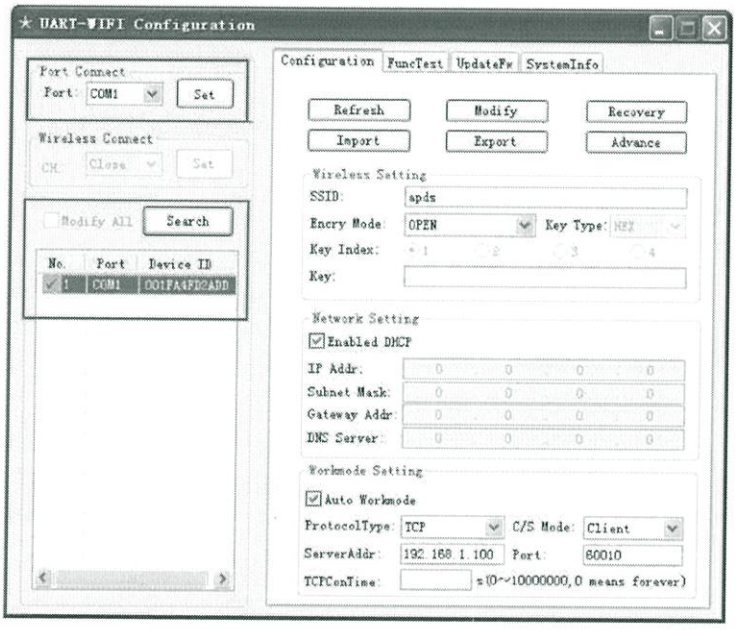
ตั้งค่าโมดูล Wireless Lan โดยต่อโมดูล Wireless Lan เข้ากับ Serial UART to USB mini B Converter V4.0 แล้วตั้งค่าผ่านโปรแกรมของผู้ผลิตที่ให้มา โดยตั้งชื่อ SSID เป็น "IPL-102" ให้ตรงกับ Access point ที่จะเชื่อมต่อ และตั้งค่า IP Address ดังนี้

IP Addr : 192.168.1.8
Subnet Mark : 255.255.255.0
Gateway Addr : 192.168.1.2
DNS Server : 192.168.1.2

โปโตคอลในการส่งข้อมูลคือ TCP Mode Client โดยส่งไปที่ Server Addr: 192.168.1.123 Port: 9092 ซึ่งคือ IP ของเครื่อง PC ที่ใช้รับข้อมูล ดังรูป



รูปที่ 3.11 การตั้งค่าโมดูล Wireless Lan ผ่าน PC



รูปที่ 3.12 โปรแกรมตั้งค่าโมดูล Wireless Lan ผ่าน PC

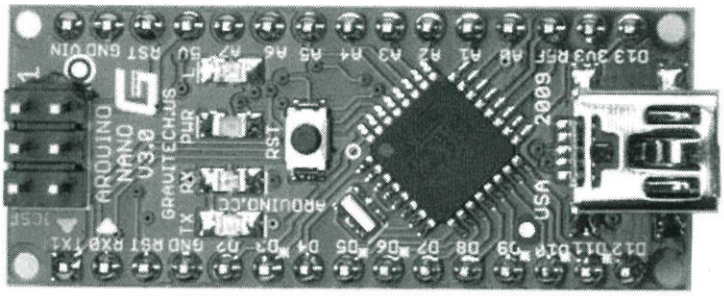
3.1.5 หลักการทำงานของไมโครคอนโทรลเลอร์ Arduino Nano 3.0

ไมโครคอนโทรลเลอร์ตัวนี้มีจุดเด่นที่ ขนาดเล็ก มีทรัพยากรพื้นฐานครบ และทำงานร่วมกับ MCU : AVR Atmega328P มีความสามารถที่เพียงพอต่อการทำงานทั่วไป

ตารางที่ 3.7 คุณสมบัติพื้นฐานของ Arduino Nano 3.0

รายการ	รายละเอียด
Microcontroller	ATmega328
Operating Voltage (Logic Level)	5 V
Input Voltage (Recommended)	7-12 V
Input Voltage (Limits)	6-20 V
Digital I/O Pins	14 (of Which 6 Provide PWM Output)
Analog Input Pins	8 Pins
DC Current per I/O Pin	40 mA

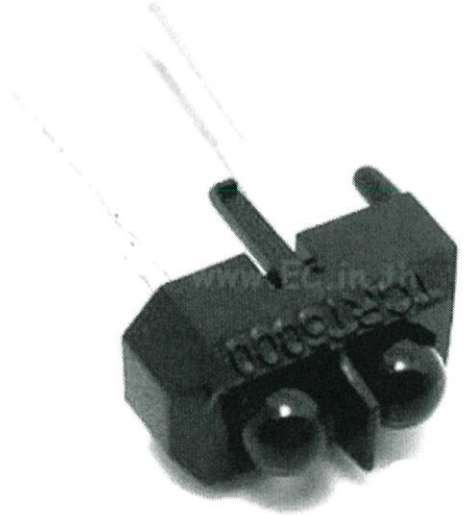
Flash Memory	32 KB (ATmega328) of which 2 KB Used by Bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"



รูปที่ 3.13 Arduino Nano 3.0 [7]

3.1.6 หลักการทำงานและคุณสมบัติของ IR Sensor

ในส่วนของ IR Sensor นั้น ใช้โมดูล TCRT5000 and TCRT5000L โดยนำมาใช้ช่วยในการวัดระยะทาง โดยจะทำให้มีค่าที่เที่ยงตรงแม่นยำมากขึ้น โดยใช้หลักการสะท้อนแสงของอินฟาเรดมาเป็นกระบวนการทำงาน



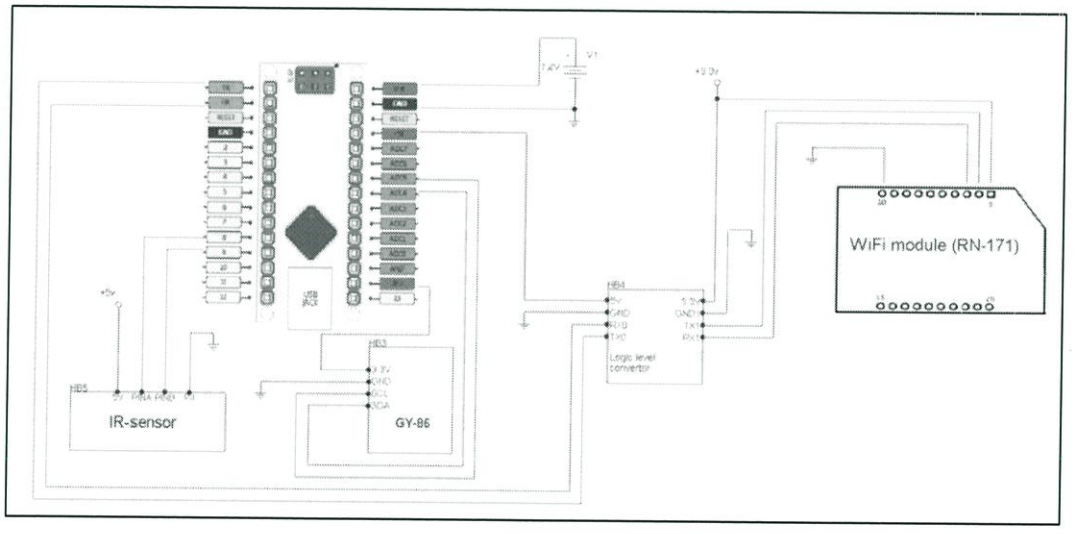
รูปที่ 3.14 IR Sensor โมดูล TCRT5000 and TCRT5000L [17]

ตารางที่ 3.8 คุณสมบัติพื้นฐานของ IR Sensor โมดูล TCRT5000 and TCRT5000L

รายการ	รายละเอียด
Package Type	Leaded
Detector Type	Phototransistor
Dimensions (L x W x H in mm)	10.2 x 5.8 x 7
Operating Range within	> 20 % Relative Collector Curren
Sensing Distance	0.2 mm to 15 mm
Typical Output	1 mA
Filter	Daylight Blocking Filter
Emitter Wavelength	950 nm

3.1.7 การต่อโมดูลต่างๆเข้าด้วยกัน และกระบวนการทำงาน

การต่อวงจรของชุดเซนเซอร์ คอนโทรลเลอร์และตัวส่งสัญญาณไร้สายนั้น จะใช้แหล่งพลังงานจากแบตเตอรี่ 2 เซลล์ เซลล์ละ 3.7v 220mAh ทำให้ได้ แรงดันไฟ 7.4v 220mAh เป็นแหล่งพลังงานให้กับวงจร และมีลายวงจรตามรูปภาพที่ 3.12



รูปที่ 3.15 ลายวงจรของอุปกรณ์ที่พัฒนาขึ้น

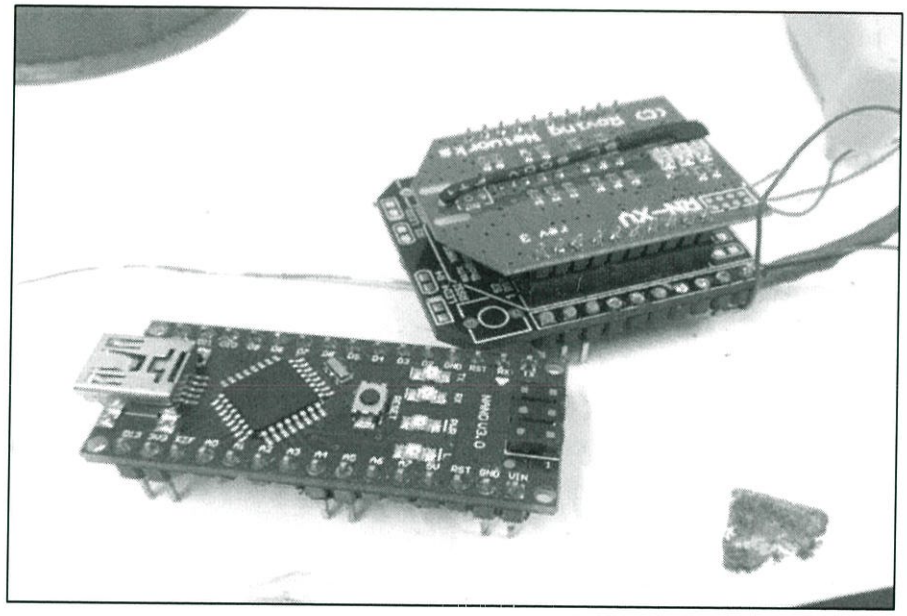
อธิบายวงจร

1. ต่อไฟเลี้ยง 3.3V ให้กับบอร์ดเซนเซอร์ GY-86 และต่อกราวน์
2. ต่อขา A4 ของ Arduino กับ SDA ของGY-86 และขา A5 กับ SCL
3. ต่อขา Tx จาก Arduino ไปที่ RX0 ของ Logic Level Converter
4. ต่อขา Rx จาก Arduino ไปที่ TX0 ของ Logic Level Converter
5. ต่อขา D8 ไปที่ PinA ของ IR-sensor
6. ต่อขา D9 ไปที่ PinB จ่ายไฟเลี้ยง 5v และต่อกราวน์ให้กับ IR-Sensor

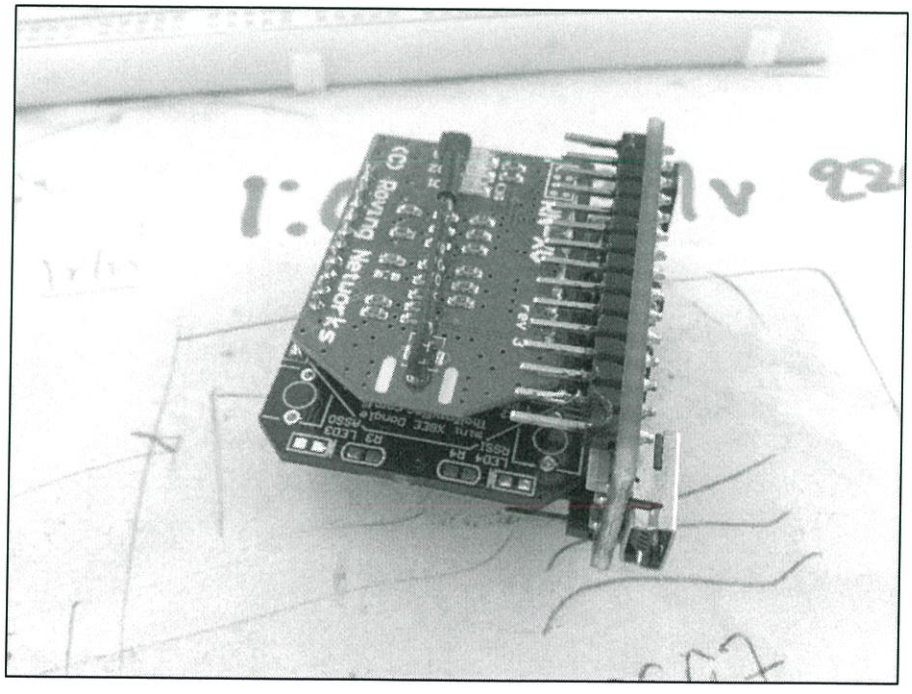
7. ต่อไฟเลี้ยง 5V และ กราวน์ จาก Arduino ไปที่ขา 5v ของ Logic Level Converter และต่อ 3.3v จาก Arduino ไปที่ขา 3.3v ของ Logic Level Converter และขาที่ 1 ของบอร์ด Wifly

8. ต่อขาที่ 2 และ 3 ของ Wifly ไปที่ TX1 และ RX1 ของ Logic Level Converter ตามลำดับ

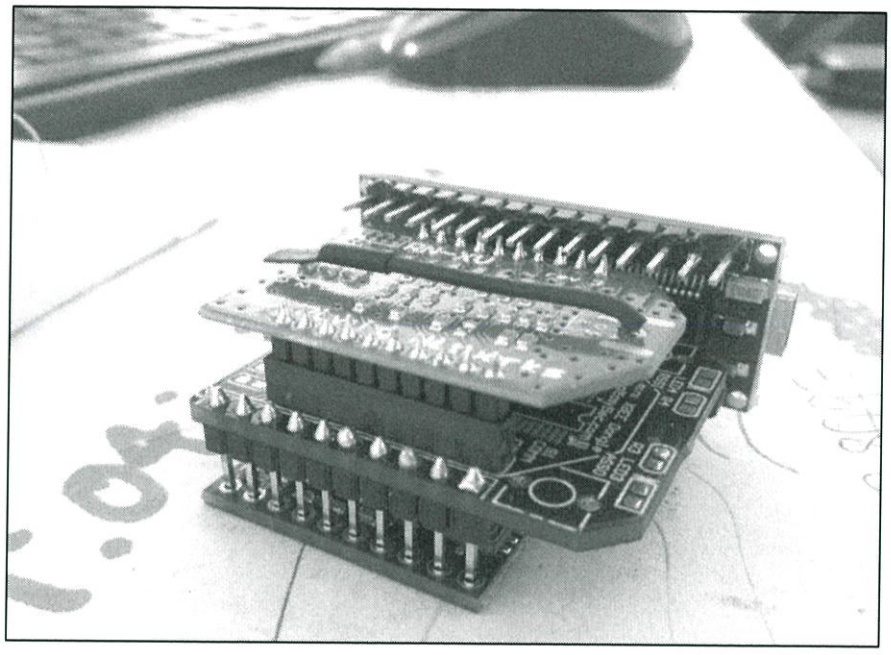
9. ต่อไฟเลี้ยงจากแบตเตอรี่เข้า ขา VIN และ GND ของ Arduino



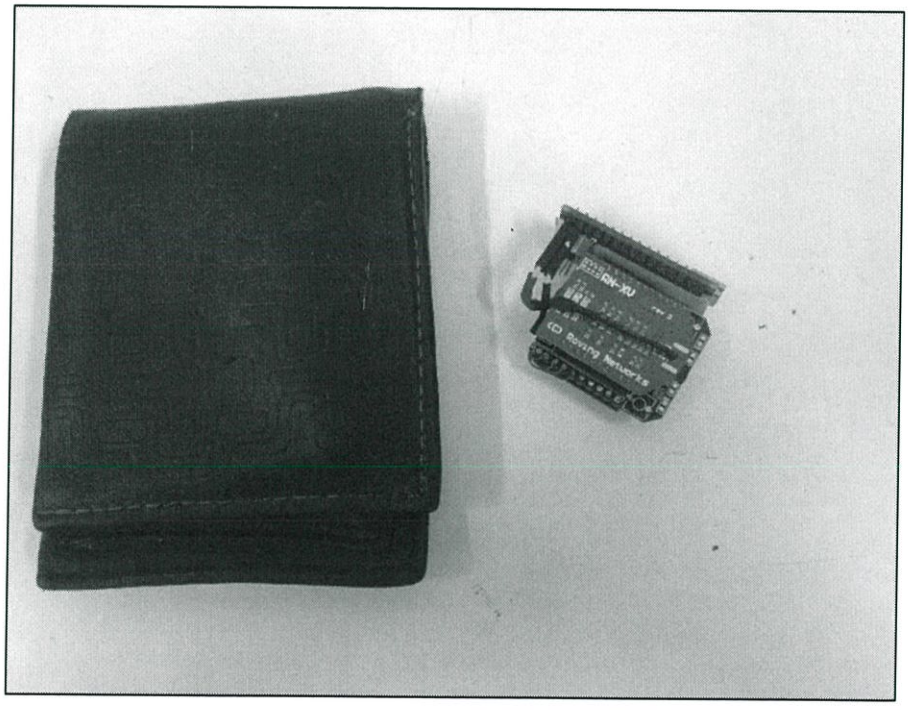
รูปที่ 3.16 การประกอบชิ้นส่วนอุปกรณ์



รูปที่ 3.17 การประกอบชิ้นส่วนอุปกรณ์



รูปที่ 3.18 การประกอบชิ้นส่วนอุปกรณ์

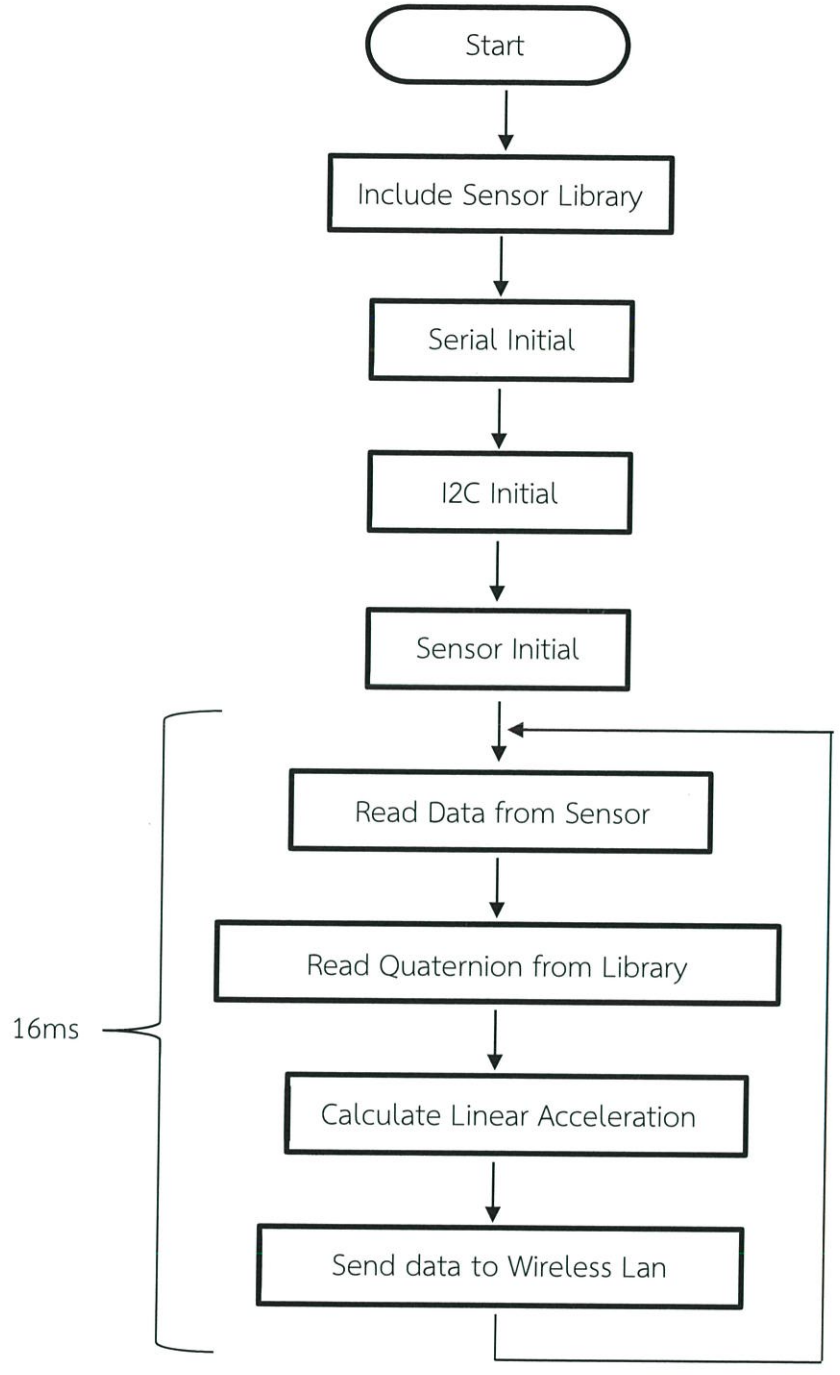


รูปที่ 3.19 ขนาดของอุปกรณ์เมื่อต่อทุกชิ้นส่วนเข้าด้วยกัน เทียบกับกระเป๋าสตางค์

กระบวนการทำงาน

ตัว Arduino อ่านค่าความเร่งจากโมดูลเซนเซอร์ Accelerometer ค่าความเร็วเชิงมุมจาก Gyro และทิศทางจาก Digital Compass โดยการอ่านค่าจากเซนเซอร์ 1 ครั้ง ใช้เวลา 16 ms (มิลลิวินาที) แล้วส่งข้อมูลการผ่านทาง Wireless Lan มาที่คอมพิวเตอร์ คอมพิวเตอร์จะอ่านค่าที่ได้จาก Wireless Lan แล้ว นำมาคำนวณหาการเคลื่อนไหว การเคลื่อนที่ของวัตถุ จากนั้นนำมาจำลองและแสดงผลรูปแบบ สามมิติ ในโปรแกรมแสดงผลที่พัฒนาขึ้นเอง โดยมีกระบวนการทำงานดังรูป

รูปที่ 3.20 กระบวนการทำงานโดยรวม



รูปที่ 3.21 Flow chart โปรแกรมของ Microcontroller Arduino Nano

3.2 การออกแบบโปรแกรม

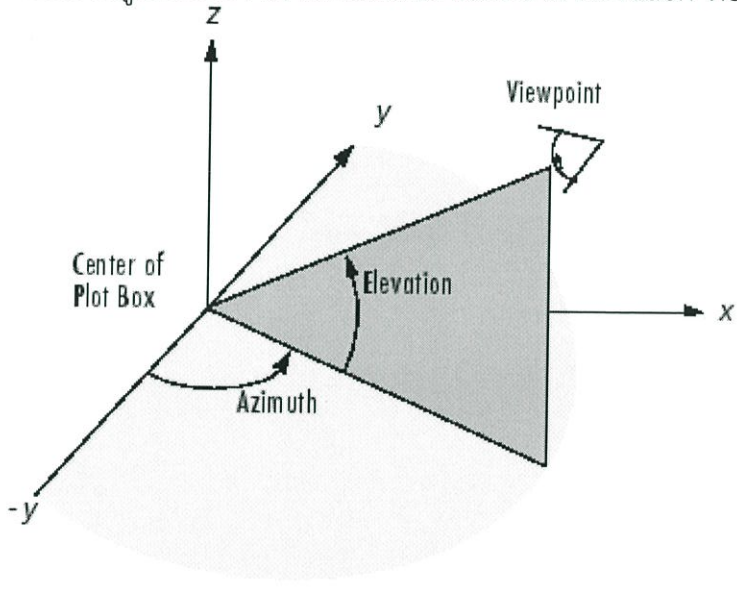
3.2.1 การออกแบบโปรแกรมเพื่อจำลองกราฟแบบ 3D

การออกแบบโปรแกรมเพื่อจำลองกราฟแบบ สามมิติ (3D) สามารถทำได้ 2 ลักษณะคือ

1. แปลงพิกัดสามมิติ x,y,z ลงในพิกัดแบบ สองมิติ (2D) โดยไม่ใช้ไลบรารี จากภายนอก
2. ใช้ไลบรารี 3D_tools ซึ่งเป็นไลบรารี มาตราฐานช่วยในการวาดรูปทรง และจำลองกราฟในรูปแบบสามมิติ

3.2.1.1 การวาดกราฟสามมิติ โดยไม่อาศัยไลบรารีจากภายนอก (3D Charts Without the WPF 3D Engine)

ในการใช้ 3D Coordinate System อาศัยการแปลงพิกัด x,y,z ลงบนพิกัด x,y โดยอาศัยกระบวนการ Projection วิธีการที่ใช้คือการคำนวณหา มุม Azimuth และ Elevation เพื่อนำมาวาดกราฟแบบ สามมิติ ในรูปแบบของ สองมิติ เป็น Azimuth and Elevation View



รูปที่ 3.22 มุม Azimuth และ Elevation

Azimuth คือ มุมที่อยู่ใน x-y Plane มีมุมที่เป็นบวกแสดงให้เห็นการหมุนทวนเข็มนาฬิกาของมุมมอง

Elevation คือ มุมที่อยู่เหนือ หรือ ต่ำกว่า x-y Plane

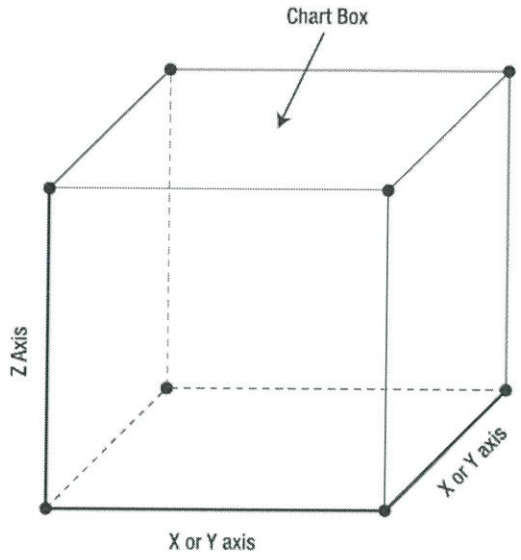
$$R_z(-\varphi)R_x(\theta-\pi/2)=\begin{pmatrix}\cos\varphi & -\sin\varphi & 0 & 0\\ \sin\varphi & \cos\varphi & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & \sin\theta & -\cos\theta & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}\cos\varphi & -\sin\varphi\sin\theta & \sin\varphi\cos\theta & 0\\ \sin\varphi & \cos\varphi\sin\theta & -\cos\varphi\cos\theta & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

จากสมการ Rotation Matrix ทางด้านบน นำมาแปลงลงในรูปแบบของโปรแกรม

```
class Utility
    {
        public static Matrix3D AzimuthElevation(double elevation,
double azimuth)
        {
            // Make sure elevation is in the range of [-90, 90]:
            if (elevation > 90)
                elevation = 90;
            else if (elevation < -90)
                elevation = -90;
            // Make sure azimuth is in the range of [-180, 180]:
            if (azimuth > 180)
                azimuth = 180;
            else if (azimuth < -180)
                azimuth = -180;
            elevation = elevation * Math.PI / 180;
            azimuth = azimuth * Math.PI / 180;
            double sne = Math.Sin(elevation);
            double cne = Math.Cos(elevation);
            double sna = Math.Sin(azimuth);
            double cna = Math.Cos(azimuth);
            Matrix3D result = new Matrix3D
                        (cna, -sne * sna, cne * sna, 0,
                         sna, sne * cna, -cne * cna, 0,
                           0,       cne,        sne, 0,
                           0,         0,          0, 1);
            return result;
        }
    }
```

หลังจากทำการแปลงพิกัด สามมิติลง สองมิติ แล้วขั้นตอนต่อไปคือการสร้าง Coordinate ขึ้นเพื่อใช้ในการวาดกราฟโดยเริ่มจากการวาดกล่องสี่เหลี่ยมเพื่อกำหนดแกนหรือ Aixs ซึ่งประกอบไปด้วย X,Y,Z[xMin, xMax, yMin, yMax, zMin, zMax]

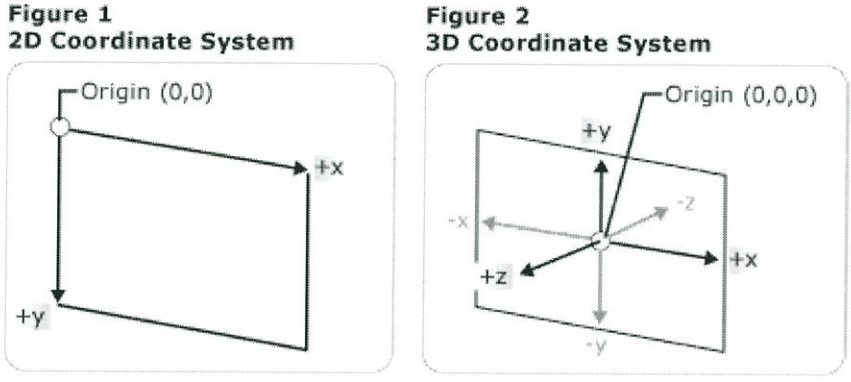


รูปที่ 3.23 Coordinate Axis [14]

และในลักษณะเดียวกับการวาด สองมิติคือต้องทำการแปลงที่อยู่ในรูปแบบพิกัดที่อยู่ในรูปแบบของ Cartesian Coordinate System ลงบน Computer Coordinate System โดยใช้หลักการเดียวกับ สองมิติ

```
public Point3D Normalize3D(Matrix3D m, Point3D pt)
        {
            Point3D result = new Point3D();
            // Normalize the point:
            double x1 = (pt.X - Xmin) / (Xmax - Xmin) - 0.5;
            double y1 = (pt.Y - Ymin) / (Ymax - Ymin) - 0.5;
            double z1 = (pt.Z - Zmin) / (Zmax - zmin) - 0.5;
            // Perform transformation on the point using matrix m:
            result.X = m.Transform(new Point3D(x1, y1, z1)).X;
            result.Y = m.Transform(new Point3D(x1, y1, z1)).Y;
            // Coordinate transformation from World to Device
system:
            double xShift = 1.05;
            double xScale = 1;
            double yShift = 1.05;
            double yScale = 0.9;
            if (Title == "No Title")
            {
                yShift = 0.95;
                yScale = 1;
            }
            if (IsColorBar)
            {
                xShift = 0.95;
                xScale = 0.9;
            }
```

```
            result.X = (xShift + xScale * result.X) * Width / 2;
            result.Y = (yShift - yScale * result.Y) * Height / 2;
            return result;
        }
```



รูปที่ 3.24 2D Coordinate System และ 3D Coordinate System [14]

ทำการเพิ่มเติมส่วนขององค์ประกอบของกราฟ เช่น Axies, Lable, xtick, ytick, ztick

```
private void AddAxes()
        {
            Matrix3D m = Utility.AzimuthElevation(Elevation,
Azimuth);
            Point3D[] pts = CoordinatesOfChartBox();
            for (int i = 0; i < pts.Length; i++)
            {
                pts[i] = Normalize3D(m, pts[i]);
            }

            DrawAxisLine(pts[0], pts[1]);
            DrawAxisLine(pts[1], pts[2]);
            DrawAxisLine(pts[2], pts[3]);
        }
```

Function ใช้ในการวาดแกน

```
private void DrawAxisLine(Point3D pt1, Point3D pt2)
        {
            Line line = new Line
            {
                X1 = pt1.X,
                Y1 = pt1.Y,
                X2 = pt2.X,
                Y2 = pt2.Y,
                Stroke = AxisColor
            };
            ChartCanvas.Children.Add(line);}
```

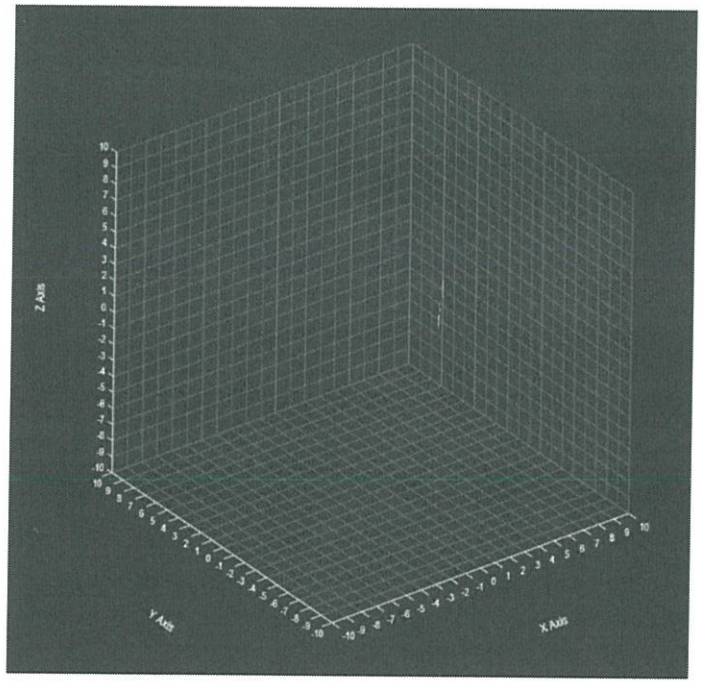
Function ใช้ในการทำ x,y,z Tick ของแต่ละแกน

```
private void AddTicks()
        {
            Matrix3D m = Utility.AzimuthElevation(Elevation,
Azimuth);
            Point3D[] pta = new Point3D[2];
            Point3D[] pts = CoordinatesOfChartBox();
            // Add x ticks:
            double offset = (Ymax - Ymin) / 30.0;
            double ticklength = offset;
            for (double x = Xmin; x <= Xmax; x = x + XTick)
            {
                if (Elevation >= 0)
                {
                    if (Azimuth >= -90 && Azimuth < 90)
                        ticklength = -offset;
                }
                else if (Elevation < 0)
                {
                    if ((Azimuth >= -180 && Azimuth < -90) ||
                    Azimuth >= 90 && Azimuth <= 180)
                        ticklength = -(Ymax - Ymin) / 30;
                }
                pta[0] = new Point3D(x, pts[1].Y + ticklength,
pts[1].Z);
                pta[1] = new Point3D(x, pts[1].Y, pts[1].Z);
                for (int i = 0; i < pta.Length; i++)
                {
                    pta[i] = Normalize3D(m, pta[i]);
                }
                AddTickLine(pta[0], pta[1]);
            }
            // Add y ticks:
            offset = (Xmax - Xmin) / 30.0;
            ticklength = offset;
            for (double y = Ymin; y <= Ymax; y = y + YTick)
            {
                pts = CoordinatesOfChartBox();
                if (Elevation >= 0)
                {
                    if (Azimuth >= -180 && Azimuth < 0)
                        ticklength = -offset;

                }
                else if (Elevation < 0)
                {
                    if (Azimuth >= 0 && Azimuth < 180)
                        ticklength = -offset;
                }
                pta[0] = new Point3D(pts[1].X + ticklength, y,
pts[1].Z);
                pta[1] = new Point3D(pts[1].X, y, pts[1].Z);
```

```
            for (int i = 0; i < pta.Length; i++)
            {
                pta[i] = Normalize3D(m, pta[i]);
            }
            AddTickLine(pta[0], pta[1]);
        }
        // Add z ticks:
        double xoffset = (Xmax - Xmin) / 45.0f;
        double yoffset = (Ymax - Ymin) / 20.0f;
        double xticklength = xoffset;
        double yticklength = yoffset;
        for (double z = Zmin; z <= Zmax; z = z + ZTick)
        {
            if (Elevation >= 0)
            {
                if (Azimuth >= -180 && Azimuth < -90)
                {
                    xticklength = 0;
                    yticklength = yoffset;
                }
                else if (Azimuth >= -90 && Azimuth < 0)
                {
                    xticklength = xoffset;
                    yticklength = 0;
                }
                else if (Azimuth >= 0 && Azimuth < 90)
                {
                    xticklength = 0;
                    yticklength = -yoffset;
                }
                else if (Azimuth >= 90 && Azimuth <= 180)
                {
                    xticklength = -xoffset;
                    yticklength = 0;
                }
            }
            else if (Elevation < 0)
            {
                if (Azimuth >= -180 && Azimuth < -90)
                {
                    yticklength = 0;
                    xticklength = xoffset;
                }
                else if (Azimuth >= -90 && Azimuth < 0)
                {
                    yticklength = -yoffset;
                    xticklength = 0;
                }
                else if (Azimuth >= 0 && Azimuth < 90)
                {
                    yticklength = 0;
                    xticklength = -xoffset;
                }
                else if (Azimuth >= 90 && Azimuth <= 180)
```

```
                    {
                        yticklength = yoffset;
                        xticklength = 0;
                    }
                }
                pta[0] = new Point3D(pts[2].X, pts[2].Y, z);
                pta[1] = new Point3D(pts[2].X + yticklength,
pts[2].Y + xticklength, z);
                for (int i = 0; i < pta.Length; i++)
                {
                    pta[i] = Normalize3D(m, pta[i]);
                }
                AddTickLine(pta[0], pta[1]);

            }

        }
```

จากโค้ดทั้งหมด จะทำให้ได้ดังรูปที่ 3.25



รูปที่ 3.25 โปรแกรมจำลอง 3D Coordinate System Without 3D Engine

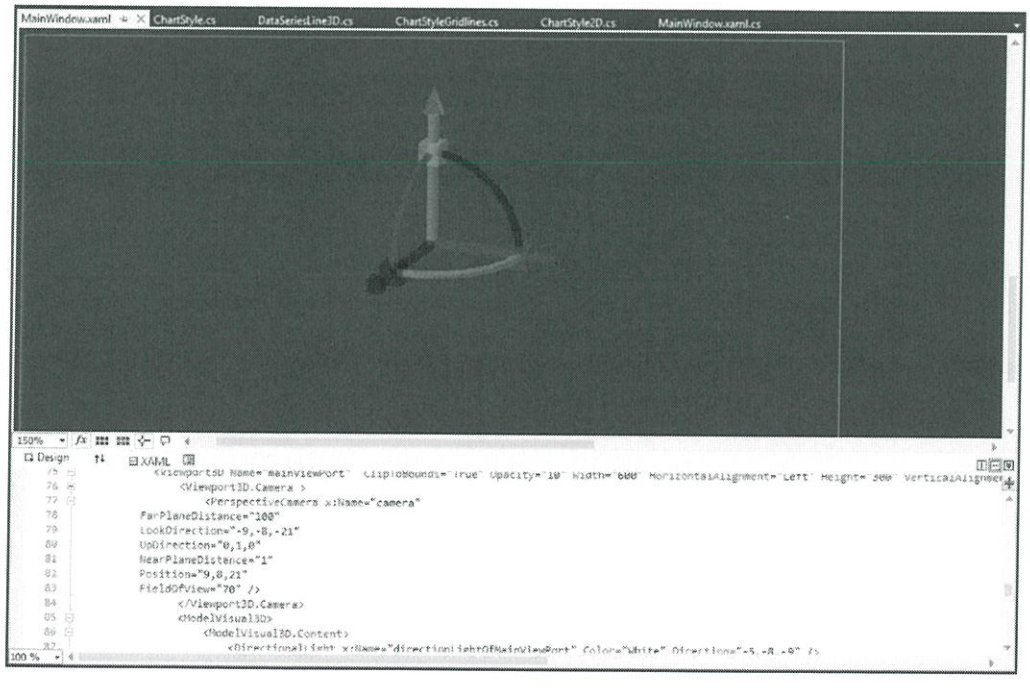
3.2.1.2 การจำลองภาพ สามมิติโดยอาศัยไลบรารี 3D_Tools

ในส่วนของการใช้ไลบรารี เพิ่มเติมจากภายนอกนั้น ด้วยความเป็นจริงแล้ว 3D Engine ของ WPF ที่มากับ .NET Framework 4.5 หรือต่ำกว่า สามารถใช้ได้ในระดับหนึ่ง เนื่องจากไลบรารี ที่เป็นพื้นฐานนำมาเพื่อจำลองโมเดลเบื้องต้น ไม่ได้ถูกออกแบบมาเพื่อ

Game Developer จึงจำเป็นต้องอาศัย 3D_Tools ซึ่งแต่ก่อนเคยเป็นไลบรารี พื้นฐานของ Framework เพื่อมาช่วยในการวาดกราฟแบบ สามมิติในส่วนที่ WPF ไม่สามารถทำได้

ขั้นตอนแรกทำการเตรียม XAML สำหรับวาดวัตถุลงบน Viewport3D ดังแสดงในรูปที่ 3.26

```
<Viewport3D Name="mainViewPort"  ClipToBounds="True"
Opacity="10" Width="600" HorizontalAlignment="Left"
Height="300" VerticalAlignment="Top">
                <Viewport3D.Camera >
                    <PerspectiveCamera x:Name="camera"
          FarPlaneDistance="100"
          LookDirection="-9,-8,-21"
          UpDirection="0,1,0"
          NearPlaneDistance="1"
          Position="9,8,21"
          FieldOfView="70" />
                </Viewport3D.Camera>
                <ModelVisual3D>
                    <ModelVisual3D.Content>
                        <DirectionalLight
x:Name="directionLightOfMainViewPort" Color="White"
Direction="-5,-8,-9" />
                    </ModelVisual3D.Content>
                </ModelVisual3D>
            </Viewport3D>
```



รูปที่ 3.26 รูปเมื่อทำการเพิ่มโค้ด

ขั้นตอนต่อไปทำการวาด Model 3D เพื่อใช้จำลองการเคลื่อนไหวของวัตถุที่ต้องการโดยใช้ Function DrawCube

```
private void DrawCube(){
            Point3D p0 = new Point3D(0, 0, 0);
            Point3D p1 =new Point3D(4, 0, 0);
            Point3D p2 =new Point3D(4, 0, 6);
            Point3D p3 =new Point3D(0, 0, 6);
            Point3D p4 =new Point3D(0, 2, 0);
            Point3D p5 =new Point3D(4, 2, 0);
            Point3D p6 =new Point3D(4, 2, 6);
            Point3D p7 = new Point3D(0, 2, 6);
            //front side triangles
            cube.Children.Add(CreateTriangleModel(p3, p2, p6));
            cube.Children.Add(CreateTriangleModel(p3, p6, p7));
            //right side triangles
            cube.Children.Add(CreateTriangleModel(p2, p1, p5));
            cube.Children.Add(CreateTriangleModel(p2, p5, p6));
            //back side triangles
            cube.Children.Add(CreateTriangleModel(p1, p0, p4));
            cube.Children.Add(CreateTriangleModel(p1, p4, p5));
            //left side triangles
            cube.Children.Add(CreateTriangleModel(p0, p3, p7));
            cube.Children.Add(CreateTriangleModel(p0, p7, p4));
            //top side triangles
            cube.Children.Add(CreateTriangleModel(p7, p6, p5));
            cube.Children.Add(CreateTriangleModel(p7, p5, p4));
            //bottom side triangles
            cube.Children.Add(CreateTriangleModel(p2, p3, p0));
            cube.Children.Add(CreateTriangleModel(p2, p0, p1));


            model.Content = cube;
            model.Transform = new Transform3DGroup();

            this.mainViewPort.Children.Add(model);
        }
```

เพิ่มเติม Function เพื่อใช้ในการวาดแกนของกราฟ

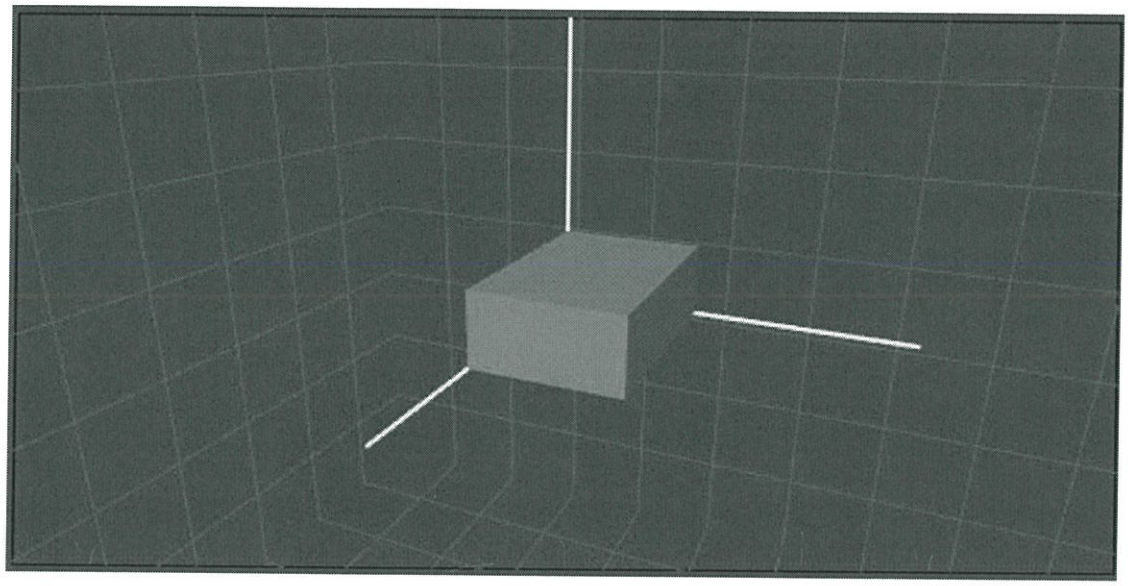
```
public void InitXYZ_Line()
        {

            Rectangle chartRect = new Rectangle();
            chartRect.Stroke = Brushes.Black;
            chartRect.StrokeThickness = 1;
            chartRect.Width = 600;
            chartRect.Height = 300;
            this.Boder.Children.Add(chartRect);
```

```
            ScreenSpaceLines3D LineX = new ScreenSpaceLines3D();
            ScreenSpaceLines3D LineY = new ScreenSpaceLines3D();
            ScreenSpaceLines3D LineZ = new ScreenSpaceLines3D();

            LineX.Thickness = LineY.Thickness = LineZ.Thickness =
2;
            LineX.Color = Colors.White;
            LineY.Color = Colors.White;
            LineZ.Color = Colors.White;
            LineX.Points.Add(new Point3D(0, 0, 0));
            LineX.Points.Add(new Point3D(10, 0, 0));
            LineY.Points.Add(new Point3D(0, 0, 0));
            LineY.Points.Add(new Point3D(0, 8, 0));
            LineZ.Points.Add(new Point3D(0, 0, 0));
            LineZ.Points.Add(new Point3D(0, 0, 10));

            this.mainViewPort.Children.Add(LineX);
            this.mainViewPort.Children.Add(LineY);
            this.mainViewPort.Children.Add(LineZ);
        }
```

เพิ่มเติม Function เพื่อใช้ในการวาดพื้นหลังของกราฟ จะได้ดังรูปที่ 3.27

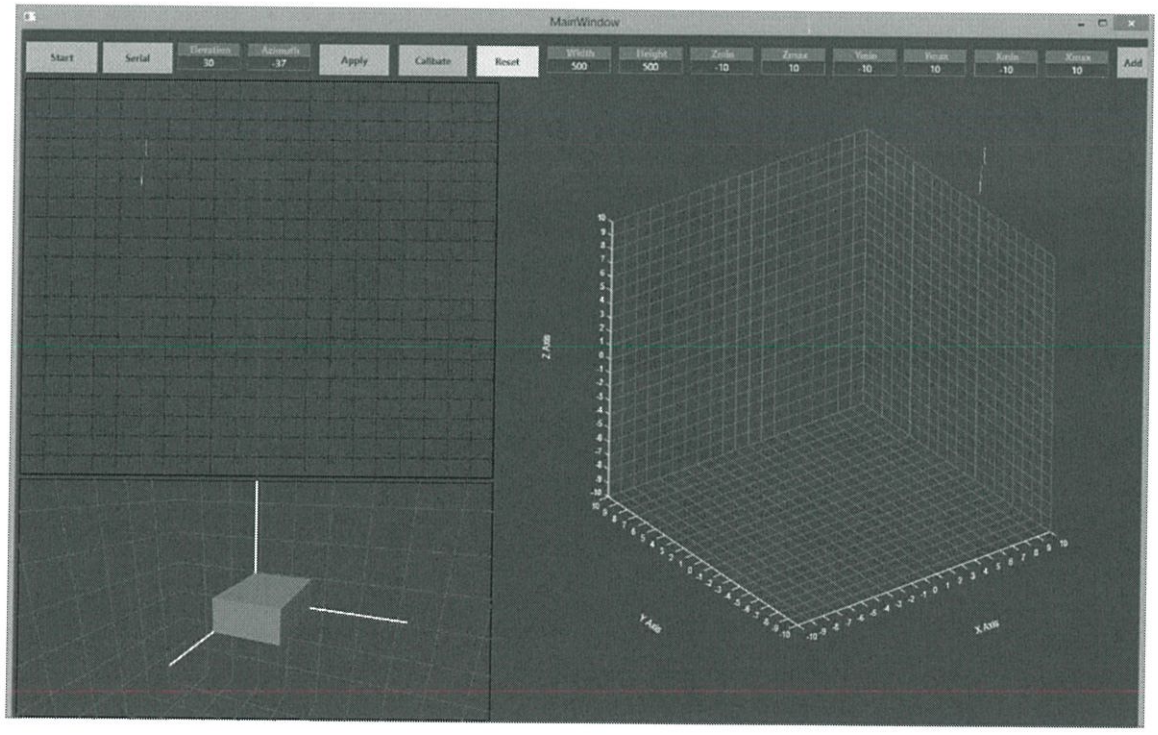
```
public void MakeLines3D()
        {
            double interval = 25;
            for (double i = -interval + 5; i <= interval; i += 5)
            {
                ScreenSpaceLines3D LinesX = new
ScreenSpaceLines3D();
                LinesX.Thickness = 0.5;
                LinesX.Color = Colors.DimGray;
                LinesX.Points.Add(new Point3D((-interval), -
interval, i * 1));
                LinesX.Points.Add(new Point3D(interval, -interval,
i * 1));

                ScreenSpaceLines3D LinesZ = new
ScreenSpaceLines3D();
                LinesZ.Thickness = 0.5;
                LinesZ.Color = Colors.DimGray;
                LinesZ.Points.Add(new Point3D(i * 1, -interval, -
interval));
                LinesZ.Points.Add(new Point3D(i * 1, -interval,
interval));


                ScreenSpaceLines3D LinesY = new
ScreenSpaceLines3D();
                LinesZ.Thickness = 0.5;
                LinesZ.Color = Colors.DimGray;
```

```
                LinesZ.Points.Add(new Point3D(-interval, i * 1, -
interval));
                LinesZ.Points.Add(new Point3D(-interval, i * 1,
interval));


                LinesZ.Thickness = 0.5;
                LinesZ.Color = Colors.DimGray;
                LinesZ.Points.Add(new Point3D(-interval, i * 1, -
interval));
                LinesZ.Points.Add(new Point3D(interval, i * 1, -
interval));


                LinesX.Thickness = 0.5;
                LinesX.Color = Colors.DimGray;
                LinesX.Points.Add(new Point3D((-interval),
interval, i * 1));
                LinesX.Points.Add(new Point3D(-interval, -interval,
i * 1));


                LinesZ.Thickness = 0.5;
                LinesZ.Color = Colors.DimGray;
                LinesZ.Points.Add(new Point3D(i * 1, interval, -
interval));
                LinesZ.Points.Add(new Point3D(i * 1, -interval, -
interval));

                mainViewPort.Children.Add(LinesX);
                mainViewPort.Children.Add(LinesZ);
            }
        }
```



รูปที่ 3.27 แบบจำลอง 3D

เมื่อนำโปรแกรมจากหัวข้อ 3.1.1.1 และ 3.1.1.2 มาประกอบกันทั้ง 2D 3D และแบบจำลอง 3D จะได้ดังรูปที่ 3.28



รูปที่ 3.28 โปรแกรมใช้ในการทดลอง

3.2.2 การเอียงวัตถุ สามมิติ

มุมที่ใช้ในการหมุนใน Viewport3D ไม่สามารถใช้มุมที่ได้จากเซนเซอร์ ได้ เนื่องจาก Viewport3D ไม่ได้การกำหนดไว้ว่าจุดไหนเป็นมุมเท่าไร และวัตถุมีการเคลื่อนไหวตลอดเวลา ทำให้ค่ามุมเปลี่ยนตลอดเวลา จึงต้องใช้ค่าก่อนหน้าเป็นค่าอ้างอิง ทำให้จำเป็นต้องนำค่าจากเซนเซอร์มาคำนวณในสมการก่อน จากนั้นจึงนำไปเขียนในโปรแกรม

ตัวอย่างการคำนวณ(1)

มุมที่ได้จากเซนเซอน์เท่ากับ 30 องศา

มุมของวัตถุปัจจุบันเท่ากับ 0 องศา

มุมที่ใช้ในการวาด = มุมที่ได้จากเซนเซอร์ - มุมของวัตถุปัจจุบัน

= 30 - 0

= 30 องศา

มุมปัจจุบันจะมีค่าเท่ากับ = มุมของวัตถุปัจจุบัน + มุมที่ใช้ในการวาด

= 0 + 30

= 30 องศา (เพื่อใช้ในการคำนวณรอบต่อไป)

ตัวอย่างการคำนวณ(2)

มุมที่ได้จากเซนเซอร์ เท่ากับ 60 องศา

มุมของวัตถุปัจจุบัน 30 องศา

มุมที่ใช้ในการวาด = มุมที่ได้จากเซนเซอร์ – มุมของวัตถุปัจจุบัน

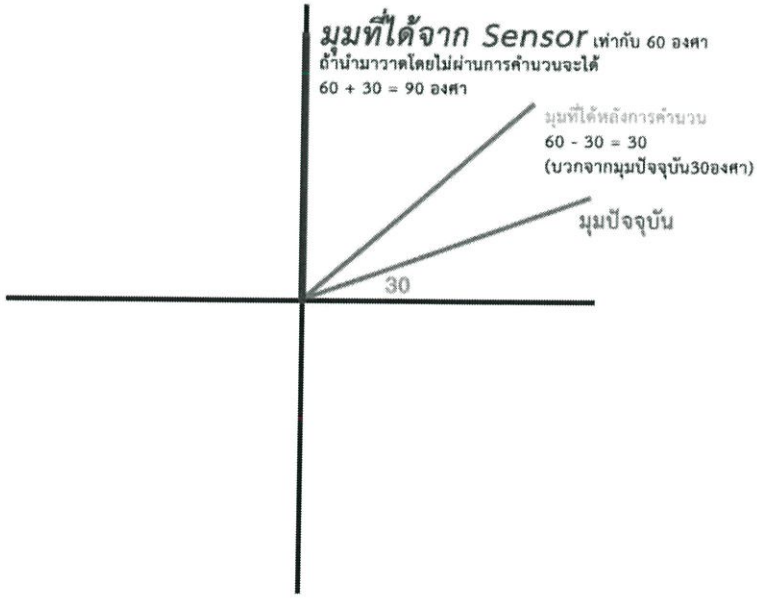
= 60 - 30

= 30 องศา

มุมปัจจุบันจะมีค่าเท่ากับ = มุมของวัตถุปัจจุบัน + มุมที่ใช้ในการวาด

= 30 + 30

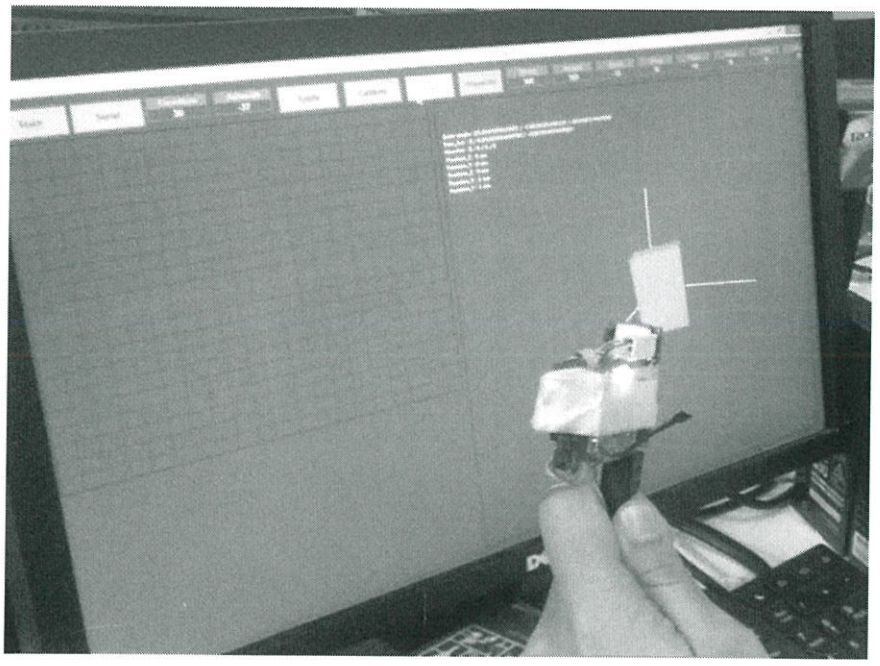
= 60 องศา (เพื่อใช้ในการคำนวณรอบต่อไป)



รูปที่ 3.29 การคำนวณมุมเพื่อใช้ในการวาดที่ถูกต้อง

นำกระบวนการที่ได้มาออกแบบโปรแกรมเพื่อใช้ในการเอียงวัตถุ ตามตัวอย่างด้านล่าง ได้ดังรูปที่ 3.30

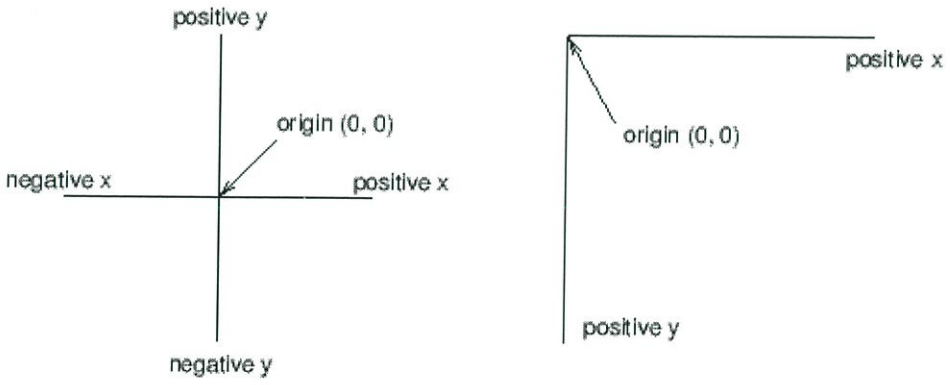
```
private void TransformWireFrameCube(double anglex, double angley,
double anglez)
        {

            //center = GetCenter(model);
            center = Center();

            double difference = anglex_Old - anglex;

            Transform3DGroup group = model.Transform as
Transform3DGroup;
            //Transform3DGroup group = wireFrameCube.Transform as
Transform3DGroup;
            AxisAngleRotation3D a = new AxisAngleRotation3D();
            a.Axis = new Vector3D(0, 0, 1);
            a.Angle = difference;
            RotateTransform3D r1 = new RotateTransform3D(a,
center);
            group.Children.Add(r1);
            anglex_Old = anglex;

            double difference2 = angley_Old - angley;

           Transform3DGroup group2 = model.Transform as
Transform3DGroup;
           //Transform3DGroup group2 = wireFrameCube.Transform as
Transform3DGroup;
           AxisAngleRotation3D a2 = new AxisAngleRotation3D();
           a2.Axis = new Vector3D(1, 0, 0);
           a2.Angle = difference2;
           RotateTransform3D r2 = new RotateTransform3D(a2,
center);
           group2.Children.Add(r2);
           angley_Old = angley;

           double difference3 = anglez_Old - anglez;

           Transform3DGroup group3 = model.Transform as
Transform3DGroup;
           //Transform3DGroup group2 = wireFrameCube.Transform as
Transform3DGroup;
           AxisAngleRotation3D a3 = new AxisAngleRotation3D();
           a3.Axis = new Vector3D(0, 1, 0);
           a3.Angle = difference3;
           RotateTransform3D r3 = new RotateTransform3D(a3,
center);
           group3.Children.Add(r3);
           anglez_Old = anglez;
                   }
```



รูปที่ 3.30 การเอียงตามมุมที่ได้จากเซนเซอร์ โดยผ่านการคำนวณแล้ว

3.2.3 การวาดกราฟ สองมิติ

การวาดกราฟ สองมิติลงใน WPF หรือในระบบพิกัดคาร์ทีเซียน (Cartesian Coordinate System) ของคอมพิวเตอร์จะแตกต่างออกไปจากระบบพิกัด คือ จุด(0,0) หรือ Origin จะอยู่ที่มุมบนซ้ายของจอแสดงผลเสมอ ดังรูปที่ 3.31



รูปที่ 3.31 ด้านซ้ายคือ Cartesian Coordinate System ด้านขวาคือ Computer Coordinate System [14]

ดังนั้นเมื่อต้องการที่จะทำการวาดกราฟลงบน พิกัดของคอมพิวเตอร์จึงจำเป็นต้องแปลงจาก จุดที่ได้จากพิกัดคาร์ทีเซียน แปลงไปเป็นพิกัดในคอมพิวเตอร์

```
private Point Normalized_Point(double Xmin, double Xmax, double Ymin, double Ymax, Point pt, Canvas canvas)
{
    Point result = new Point();
    result.X = (pt.X - Xmin) * canvas.Width / (Xmax - Xmin);
    result.Y = canvas.Height - (pt.Y - Ymin) * canvas.Height / (Ymax - Ymin);
    return result;
}
```

องค์ประกอบของการวาดกราฟ สองมิติ

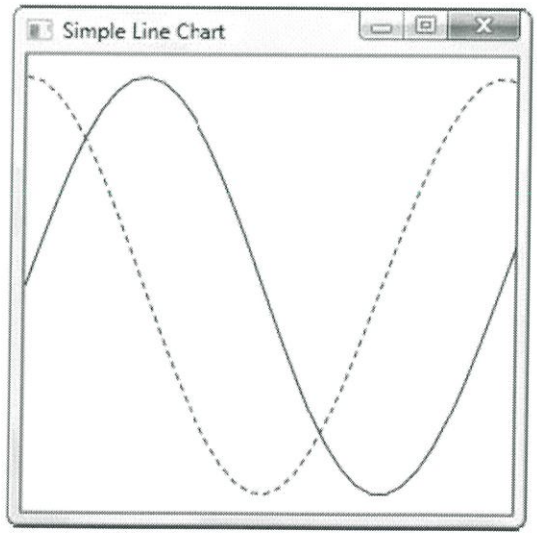
1. Axis แกนของกราฟ เช่น x, y
2. Line เส้นที่ใช้ในการวาดกราฟ
3. Text ข้อความที่ใช้แสดง
4. Title ข้อความที่บอกว่ากราฟนี้คือกราฟอะไร
5. Label ใช้บอกว่าแกนนี้คือแกนอะไร
6. Legend ใช้บอกคุณสมบัติของ Line ที่ใช้ในการวาด

ก่อนทำการวาดกราฟจำเป็นต้องกำหนดรูปแบบของ XAML ก่อนเพื่อใช้ทำการวาดต่อไป เพื่อให้ Class อื่นๆทำการอ้างอิง

```
<Window x:Class="LineCharts.SimpleLineChart"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Simple Line Chart" Height="300" Width="300">
    <Grid  Margin="0" x:Name ="chartGrid" ClipToBounds="True"
           Background="Transparent" SizeChanged="chartGrid_SizeChanged">
        <Border HorizontalAlignment="Center" VerticalAlignment="Center"
                BorderBrush="Gray" BorderThickness="1"
                Background="White" Margin="0">
            <Canvas  Margin="0" x:Name ="chartCanvas" ClipToBounds="True"
                     Background="Transparent"/>
        </Border>
    </Grid>
</Window>
```

เขียน Function เพื่อทดลองวาดกราฟตัวอย่างกราฟที่วาดคือกราฟ Cosines และ Sin ดังรูปที่ 3.32

```
private void AddChart()
{
    // Draw sine curve:
    pl = new Polyline();
    pl.Stroke = Brushes.Black;
    for (int i = 0; i < 70; i++)
    {
        double x = i/5.0;
        double y = Math.Sin(x);
        pl.Points.Add(NormalizePoint(new Point(x, y)));
    }
    chartCanvas.Children.Add(pl);


    // Draw cosine curve:
    pl = new Polyline();
    pl.Stroke = Brushes.Black;
    pl.StrokeDashArray = new DoubleCollection(new double[] { 4, 3 });

    for (int i = 0; i < 70; i++)
    {
        double x = i / 5.0;
        double y = Math.Cos(x);
        pl.Points.Add(NormalizePoint(new Point(x, y)));
    }
    chartCanvas.Children.Add(pl);
}
```

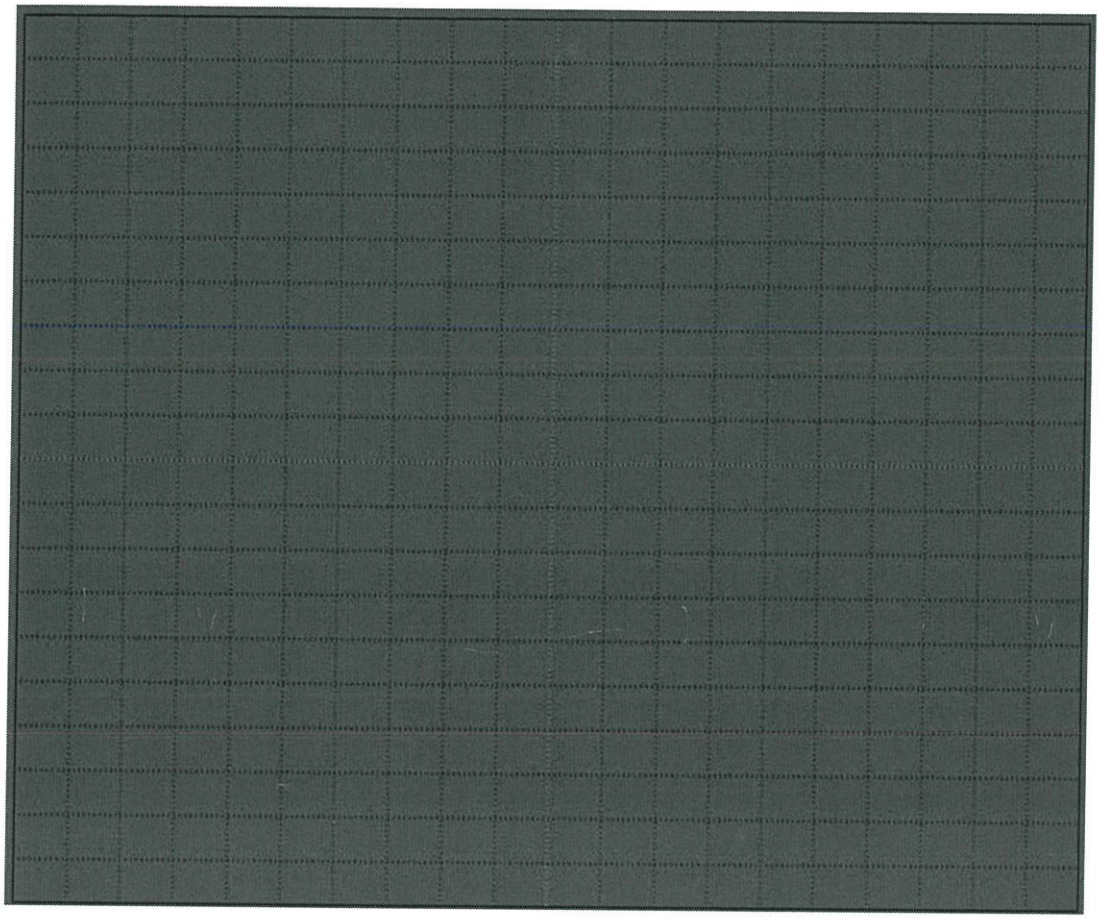


รูปที่ 3.32 รูปเมื่อทำการรันโปรแกรม

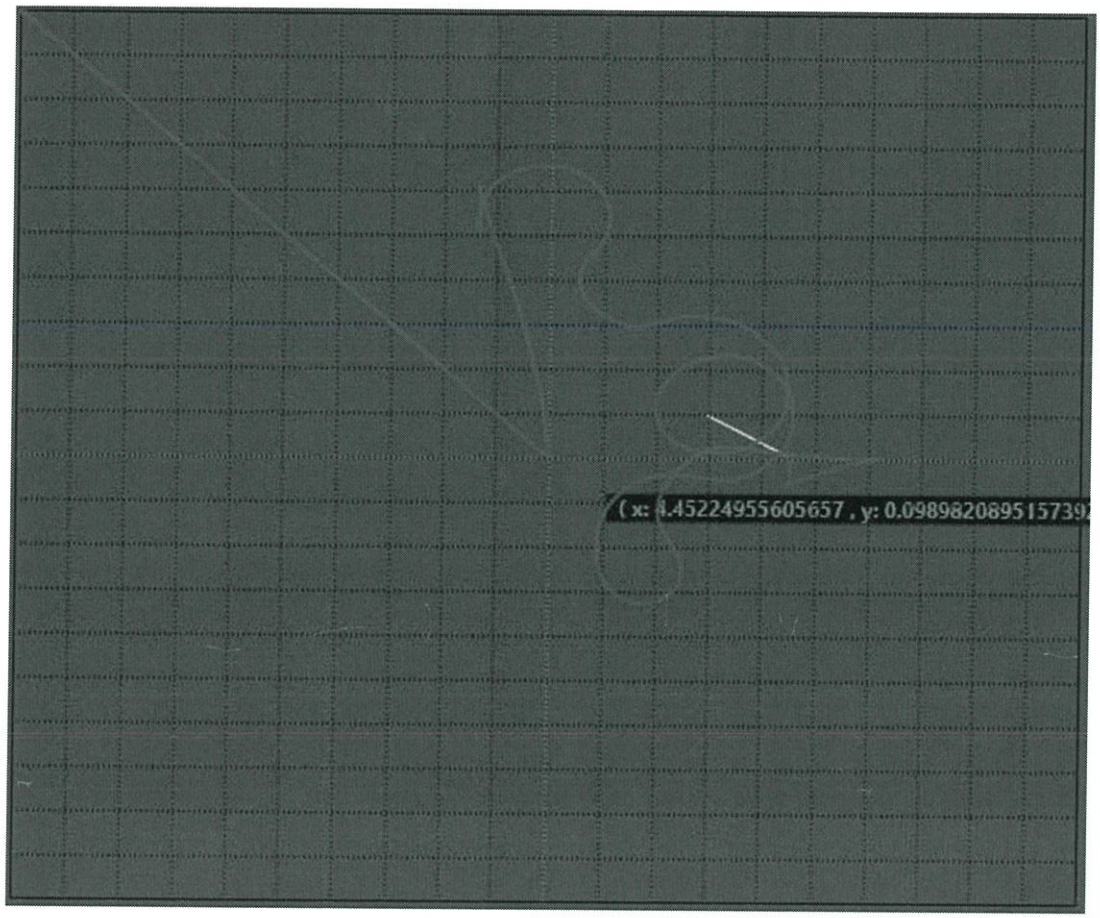
ทำการปรับปรุงเพื่อนำไปใช้งานจริงโดยการเพิ่ม xTick, yTick เพื่อทำการวาดเส้นเป็นช่วงๆ เพื่อบอกพิกัดในแต่ละส่วน จะได้ดังรูปที่ 3.33 และ 3.34

```
//See all this code in Reference Chapter
public void AddChartStyle()
        {
            Point pt = new Point();
            Line tick = new Line();
            double offset = 0;
            double dx, dy;
            TextBlock tb = new TextBlock();
            // determine right offset:
            tb.Text = Xmax.ToString();
            tb.Measure(new Size(Double.PositiveInfinity,
Double.PositiveInfinity));
            Size size = tb.DesiredSize;
            rightOffset = size.Width / 2 + 2;
            // Determine left offset:
            for (dy = Ymin; dy <= Ymax; dy += YTick)
            {
            pt = NormalizePoint(new Point(Xmin, dy));
            tb = new TextBlock();
            tb.Text = dy.ToString();
            tb.TextAlignment = TextAlignment.Right;
            tb.Measure(new Size(Double.PositiveInfinity,
            Double.PositiveInfinity));
            size = tb.DesiredSize;
            if (offset < size.Width)
            offset = size.Width;
            }
            leftOffset = offset + 5;
            Canvas.SetLeft(ChartCanvas, leftOffset);
            Canvas.SetBottom(ChartCanvas, bottomOffset);

            Rectangle chartRect = new Rectangle();
            chartRect.Stroke = Brushes.Black;
            chartRect.Width = ChartCanvas.Width;
            chartRect.Height = ChartCanvas.Height;


        ChartCanvas.Children.Add(chartRect);

            // Create vertical gridlines:
            if (IsYGrid == true)
            {
                for (dx = Xmin + XTick; dx < Xmax; dx += XTick)
            {
            gridline = new Line();
            AddLinePattern();
            gridline.X1 = NormalizePoint(new Point(dx, Ymin)).X;
            gridline.Y1 = NormalizePoint(new Point(dx, Ymin)).Y;
            gridline.X2 = NormalizePoint(new Point(dx, Ymax)).X;
            gridline.Y2 = NormalizePoint(new Point(dx, Ymax)).Y;
```

```
            ChartCanvas.Children.Add(gridline);
            }
            }
            // Create horizontal gridlines:
            if (IsXGrid == true)
            {
                for (dy = Ymin + YTick; dy < Ymax; dy += YTick)
                {
                    gridline = new Line();
                    AddLinePattern();
                    gridline.X1 = NormalizePoint(new Point(Xmin,
                    dy)).X;
                    gridline.Y1 = NormalizePoint(new Point(Xmin,
                    dy)).Y;
                    gridline.X2 = NormalizePoint(new Point(Xmax,
                    dy)).X;
                    gridline.Y2 = NormalizePoint(new Point(Xmax,
                    dy)).Y;
                    ChartCanvas.Children.Add(gridline);
                }
            }
            if (IsXGrid == true)
            {
                for (dy = Ymin + YTick; dy < Ymax; dy += YTick)
                {
                    gridline = new Line();
                    AddLinePattern();
                    gridline.X1 = NormalizePoint(new Point(Xmin,
                    dy)).X;
                    gridline.Y1 = NormalizePoint(new Point(Xmin,
                    dy)).Y;
                    gridline.X2 = NormalizePoint(new Point(Xmax,
                    dy)).X;
                    gridline.Y2 = NormalizePoint(new Point(Xmax,
                    dy)).Y;
                    ChartCanvas.Children.Add(gridline);
                }
            }
        }
```



รูปที่ 3.33 หน้าจอแสดงผลเมื่อทำการเพิ่มเติม Function AddChartStyle ลงในโปรแกรม



รูปที่ 3.34 การแสดงผลของโปรแกรม เมื่อใช้งานจริง

3.2.4 การวาดเส้นลง Canvas 2D

การวาดลง Canvas โดยเพื่อแสดงการ tracking ของวัตถุโดยใช้ LineGeometry, GeometryGroup เนื่องจากการวาดเส้นเพื่อติดตามวัตถุนั้น มีการรับข้อมูลแบบ Realtime ที่มีความเร็วสูงมากประมาณ 16ms ต่อรอบการวาด ซึ่งจะใช้การวาดแบบ Startpoint to Endpoint ทำงานโดยวาดแล้วนำไป Marge กับภาพเดิม ซึ่งจะแตกต่างจากการวาดแบบที่เก็บ Point ไว้ใน List<> เนื่องจากต้องมีการนำจุดที่เคยวาดแล้ว นำมาวาดใหม่ทุกรอบที่มีการรับข้อมูล แต่ข้อเสียของ GeometryGroup นี้คือสิ้นเปลืองหน่วยความจำเป็นอย่างมาก

ในโปรแกรมมีการวาดแบบ 3D ใน Canvas แบบ 2D ดังนั้นการวาดต้องทำการแปลงโดยใช้ Function Normalize3D ซึ่งเป็น Function เดียวกับที่ใช้ในการสร้างการจำลองแบบ 3D โดยไม่อาศัยไลบรารี จากภายนอก

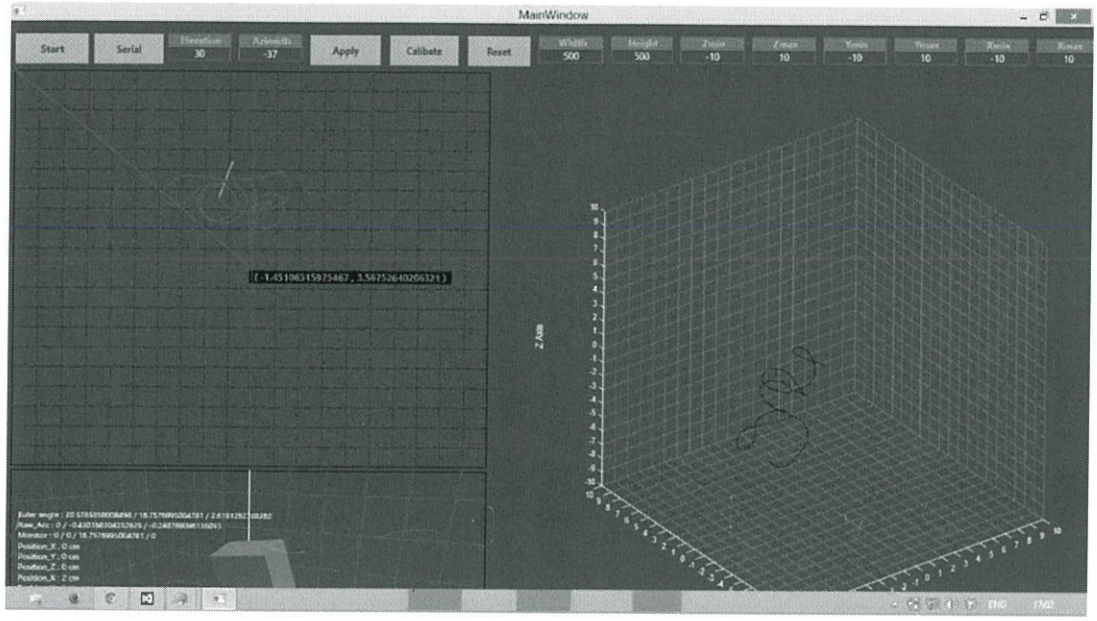
```
public Point3D Normalize3D(Matrix3D m, Point3D pt)
        {
            Point3D result = new Point3D();
            // Normalize the point:
            double x1 = (pt.X - Xmin) / (Xmax - Xmin) - 0.5;
            double y1 = (pt.Y - Ymin) / (Ymax - Ymin) - 0.5;
```

```
            double z1 = (pt.Z - Zmin) / (Zmax - zmin) - 0.5;
            // Perform transformation on the point using matrix m:
            result.X = m.Transform(new Point3D(x1, y1, z1)).X;
            result.Y = m.Transform(new Point3D(x1, y1, z1)).Y;
            // Coordinate transformation from World to Device
system:
            double xShift = 1.05;
            double xScale = 1;
            double yShift = 1.05;
            double yScale = 0.9;
            if (Title == "No Title")
            {
                yShift = 0.95;
                yScale = 1;
            }
            if (IsColorBar)
            {
                xShift = 0.95;
                xScale = 0.9;
            }
            result.X = (xShift + xScale * result.X) * Width / 2;
            result.Y = (yShift - yScale * result.Y) * Height / 2;
            return result;        }
```

ตัวอย่างโปรแกรมการวาดเส้นลงบน Canvas บนแกน สองมิติ

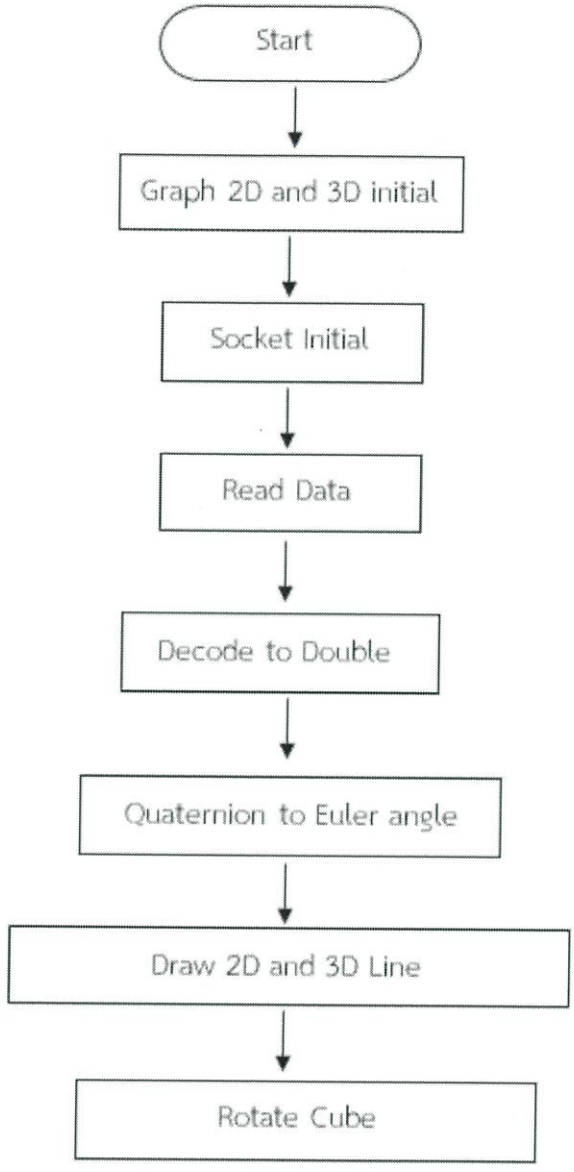
```
public void drawPoly_XY(Point pt)
        {
      pt = Normalized_Point(Convert.ToDouble(xmin.Text),
      Convert.ToDouble(xmax.Text), Convert.ToDouble(ymin.Text),
      Convert.ToDouble(ymax.Text), pt, coordinate_XY);
                        double x = pt.X;
            double y = pt.Y;
                        Point pOne = new Point(xOld_XY, yOld_XY);
            Point pTwo = new Point(x, y);
            GeometryGroup lineGroup = new GeometryGroup();
            LineGeometry connectorGeometry = new LineGeometry {
                                                    StartPoint = pOne,
                                                    EndPoint = pTwo };
            lineGroup.Children.Add(connectorGeometry);
            path = new System.Windows.Shapes.Path
            {
                Data = lineGroup,
                StrokeThickness = 2,
                Stroke = Brushes.Red,
                Fill = Brushes.Black
            };
            lineGroupDewX.Children.Add(connectorGeometry);
            coordinateSystem_XY.Children.Add(path);
                        xOld_XY = x;
            yOld_XY = y;
              }
```

```
private void DrawLine3D(Point3D pt)
        {
            Matrix3D m = Utility.AzimuthElevation(cs.Elevation,
cs.Azimuth);
            Point3D temp;
            temp = cs.Normalize3D(m, pt);

            if (first)
            {
                temp = cs.Normalize3D(m, new Point3D(xOld_XY3D,
yOld_XY3D, 0));
                xOld_XY3D = temp.X;
                yOld_XY3D = temp.Y;
                first = false;
            }

            Point pOne = new Point(xOld_XY3D, yOld_XY3D);
            Point pTwo = new Point(temp.X, temp.Y);

            GeometryGroup lineGroup = new GeometryGroup();
            LineGeometry connectorGeometry = new LineGeometry {
StartPoint = pOne, EndPoint = pTwo };

            lineGroup.Children.Add(connectorGeometry);
            path = new System.Windows.Shapes.Path
            {
                Data = lineGroup,
                StrokeThickness = 1,
                Stroke = Brushes.Black,
                Fill = Brushes.Black
            };

            lineGroupDewX.Children.Add(connectorGeometry);
            cs.ChartCanvas.Children.Add(path);
            xOld_XY3D = temp.X;
            yOld_XY3D = temp.Y;

            //PointList.Add(pOne);
              }
```



รูปที่ 3.35 การวาดกราฟ 2D (ด้านซ้าย) การวาดเส้น 3D ลงบน Canvas แบบ 2D

3.2.5 ภาพรวมการทำงานของโปรแกรม



รูปที่ 3.36 Flowchart ภาพรวมการทำงานของโปรแกรม

จากรูปที่ 3.36 ขั้นตอนทำงานของโปรแกรมเริ่มต้นโดย Graph 2D and 3D Initial คือการเตรียมพื้นที่สำหรับจำลองการ Tracking วัตถุและการจำลองการเอียงของวัตถุ โดยโปรแกรมจะเริ่มเมื่อทำการกดปุ่ม Start

```
1 reference
private void Start_Click(object sender, RoutedEventArgs e)
{
    tcpListener = new TcpListener(IPAddress.Any, 9092);
    listenThread = new Thread(new ThreadStart(ListenForClients));
    listenThread.Start();
}
```

โปรแกรมทำการเรียก Function ListenForClient โดยเรียกผ่าน Thread เพราะโปรแกรมมีการรับข้อมูลมาในลักษณะ Realtime ในหน่วย Millisecond ถ้าไม่ทำการเรียกผ่าน Thread จะทำให้โปรแกรมค้างได้

```
1 reference
private void ListenForClients()
{
    tcpListener.Start();

    while (true)
    {
        client = this.tcpListener.AcceptTcpClient();
        Thread clientThread = new Thread(new ThreadStart(HandleClientComm));
        clientThread.Start();
    }
}
```

หลังจากเรียก Function ListenForClient แล้ว ListenForClient จะทำการเรียก Function HandleClientComm เพื่อรองรับข้อมูลเข้ามาอีก เพื่อไม่ให้ข้อมูลที่เข้ามาล้น จนทำให้ข้อมูลบางส่วนหายไป

```
1 reference
public void HandleClientComm()
{
    NetworkStream clientStream = client.GetStream();
    byte[] message = new byte[4096];
    int bytesRead;

    while (true)
    {
        bytesRead = 0;
        try
        {
            bytesRead = clientStream.Read(message, 0, 4096);
        }
        catch
        {
            break;
```

```
            }
            if (bytesRead == 0)
            {
                break;

            }
            encoder = System.Text.Encoding.ASCII.GetString(message, 0, bytesRead);
            Dispatcher.BeginInvoke((Action)(() => UpdateTextbox(encoder)), null);

        }
        client.Close();
    }
```

HandleClientComm ทำการเรียก Class NetworkStream เพื่อรับค่าเข้ามา Buffer ไว้ และทำการแปลงค่าจาก byte ที่รับได้ไปเป็น String หรือข้อความ ด้วยคำสั่ง ASCII.GetString หลังจากนั้นทำการส่งค่าเพื่อไปทำการ Decode ออก เพราะค่าที่เข้ามามีการ Encode ไว้

```
1 reference
public void UpdateTextbox(string message)
{
    string[] splitData;

    splitData = message.Split(',');

    if (splitData.Length >= 10)// so we have 7 elements include \r\n
    {
        for (int i = 0; i < 9; i++) // less than upper 1 value
        {
            convertToDouble[i] = decodeFloat(splitData[i]);
        }
    }
    q[0] = convertToDouble[0];
    q[1] = convertToDouble[1];
    q[2] = convertToDouble[2];
    q[3] = convertToDouble[3];

    quaternionToEuler(q, YawPitchRoll);
    if (click_cali) quaternionToEuler(quatProd(hq, q), Euler);
    if (click_cali)
    {
        Dispatcher.BeginInvoke((Action)(() => Svt(convertToDouble[8])), null);
        monitor[0] = Euler[2];
    }
    Dispatcher.BeginInvoke((Action)(() => Trackdata()), null);
    Dispatcher.BeginInvoke((Action)(() =>
    TransformWireFrameCube(-1 * Euler[1], Euler[2], Euler[0])), null);
    monitor[2] = Euler[1];
}
```

การแยกข้อมูลที่รับมาได้จะแตกต่างจากการเขียนในโปรแกรมก่อนหน้า เนื่องจากมีการเปลี่ยนตัวส่งข้อมูลเป็นแบบ WiFi ที่มีราคาสูงกว่าทำให้มีข้อมูล ที่รับเข้ามาไม่มีข้อมูล ที่ไม่ต้องการ จึงไม่มีเพิ่มกรณีที่ตรวจว่าข้อมูล นั้นส่งมาถูกต้องหรือไม่ ประกอบด้วย Class ที่ใช้ในการรับมีการรับแบบ NetworkStream ซึ่งจะทำการรับข้อมูลเป็นบันทัดจึงไม่เกิดการ Flow ของข้อมูล หลังจากนั้นนำข้อมูลที่ได้รับมาแปลงเป็นตัวเลขเนื่องจากได้รับมาเป็น hex โดยใช้ Function decodeFloat

```
2 references
private static float decodeFloat(String instring)
{
    byte[] indata = new byte[4];
    if (instring.Length == 8)
    {
        indata[0] = (byte)Convert.ToInt32(instring.Substring(0, 2), 16);
        indata[1] = (byte)Convert.ToInt32(instring.Substring(2, 2), 16);
        indata[2] = (byte)Convert.ToInt32(instring.Substring(4, 2), 16);
        indata[3] = (byte)Convert.ToInt32(instring.Substring(6, 2), 16);
    }
    int intbits = (indata[3] << 24) | ((indata[2] & 0xff) << 16) | ((indata[1] & 0xff) << 8) | (indata[0] & 0xff);
    byte[] bytes = BitConverter.GetBytes(intbits);
    return BitConverter.ToSingle(bytes, 0);
}
```

ตัวอย่างข้อมูลที่รับมาได้ไม่มีค่าผิดเพี้ยนเมื่อเทียบกับค่าที่ได้รับจากไวไฟตัวก่อน

```
splitData = message.Split(',');
                message "D3EC6F3F,C54E1FBD,8484FCBD,E529A3BE,1DEE82BE,F934BABB,DA6676BF,B6F3FD3C,00000000,\r\n"
if (splitData.Length >= 10)// so we have 7 elements include \r\n
{
    for (int i = 0; i < 9; i++) // less than upper 1 value
    {
        convertToDouble[i] = decodeFloat(splitData[i]);
        //defult data
```

ขั้นตอนต่อไป เมื่อแปลงข้อมูลจาก hex เป็นตัวเลขหรือค่าที่ต้องการได้แล้ว ทำการแปลงค่าที่ได้ซึ่งเป็นค่า Quaternion มาแปลงเป็นข้อมูลที่ต้องการ ซึ่งค่าของ Quaternion ที่ได้รับมานั้นนำมาหาค่า มุมหรือ Euler ในหน่วยองศา และค่าที่ได้นอกเหนือจาก Quaternion คือค่าระยะทางที่ได้จาก Sensor

```
splitData = message.Split(',');

if (splitData.Length >= 10)// so we have 7 elements include \r\n
{
    for (int i = 0; i < 9; i++) // less than upper 1 value
    {
        convertToDouble[i] = decodeFloat(splitData[i]);
        //defult
        //q0
        //q1
        //q2
        //q3
        //lineae_ac
        //lineae_ac
        //lineae_ac
        //dt
    }
```

convertToDouble {double[11]}

[0]	0.93720740079879761
[1]	-0.038893479853868484
[2]	-0.12329962849617004
[3]	-0.31867900490760803
[4]	-0.25572291016578674
[5]	-0.00568258436396718
[6]	-0.962506890296936
[7]	0.030999999493360519
[8]	0.0
[9]	0.0
[10]	0.0

ตารางที่ 3.9 ตำแหน่งข้อมูลและชนิดข้อมูล

ตำแหน่งของข้อมูล	ชนิดของข้อมูล
0	Quaternion_1
1	Quaternion_2
2	Quaternion_3
3	Quaternion_4
4	Linear Accerloration X
5	Linear Accerloration Y
6	Linear Accerloration Z
7	Distance
8	n/a
9	n/a
10	n/a

ตัวอย่างโค้ด เพื่อแปลงเป็น Euler ต้องทำการ Invert ค่า Quaternion โดยใช้ Function quatProd

```
2 references
double[] quatProd(double[] a, double[] b)
{
    double[] q = new double[4];

    q[0] = a[0] * b[0] - a[1] * b[1] - a[2] * b[2] - a[3] * b[3];
    q[1] = a[0] * b[1] + a[1] * b[0] + a[2] * b[3] - a[3] * b[2];
    q[2] = a[0] * b[2] - a[1] * b[3] + a[2] * b[0] + a[3] * b[1];
    q[3] = a[0] * b[3] + a[1] * b[2] - a[2] * b[1] + a[3] * b[0];

    return q;
}
```

```
4 references
void quaternionToEuler(double[] q, double[] euler)
{
    euler[0] = (Math.Atan2(2 * q[1] * q[2] - 2 * q[0] * q[3], 2 * q[0] * q[0] + 2 * q[1] * q[1] - 1))*180.0/Math.PI; // psi
    euler[1] = (-Math.Asin(2 * q[1] * q[3] + 2 * q[0] * q[2]))*180.0/Math.PI; // theta
    euler[2] = (Math.Atan2(2 * q[2] * q[3] - 2 * q[0] * q[1], 2 * q[0] * q[0] + 2 * q[3] * q[3] - 1))*180.0/Math.PI; // phi
}
```

ตัวอย่างการเรียกใช้ Function ทั้งหมด

```
if (click_cali) quaternionToEuler(quatProd(hq, q), Euler);

if (click_cali)
{
```

ขั้นตอนต่อไปเมื่อได้ค่าข้อมูลที่ต้องการแล้วนำข้อมูลทั้งหมดมาวาดกราฟ หรือ จำลองการเอียงวัตถุ โดยใช้ Function Svt ซึ่งใน WPF แยกส่วนของ UI กับส่วนของโปรแกรมฉะนั้นเมื่อทำการส่งค่าไปวาดที่ส่วนของ UI จำเป็นต้องเพิ่มคำสั่ง Dispatcher.BeginInvoke

```
Dispatcher.BeginInvoke((Action)(() => Svt(convertToDouble[8])), null);
```

```
933         }
934         public void Svt(double speed)
935         {
936             if (Euler[0] < 0) Euler[0] =  Euler[0] + 360;
937
938             double conn= speed;
939             double x, y, z;
940
941
942             if (Euler[0] >= 90 && Euler[0] < 180.0 ) //interval fig
943             {//Q4
944
945                 y = -1*conn * Math.Sin((Euler[0]-90) * Math.PI / 180.0) + y_Position_Old;
946                 x =    conn * Math.Cos((Euler[0]-90) * Math.PI / 180.0) + x_Position_Old;
947                 x_Position_Old = x;
948                 y_Position_Old = y;
949
950             }
951             else
952             {//Q1
953                  y = conn * Math.Cos(Euler[0] * Math.PI / 180.0) + y_Position_Old;
954                  x = conn * Math.Sin(Euler[0] * Math.PI / 180.0) + x_Position_Old;
955                  x_Position_Old = x;
956                  y_Position_Old = y;
957             }
958             z = -1*conn*Math.Sin(Euler[2] * Math.PI/180.0) + z_Position_Old;
959             z_Position_Old = 0;
960             Dispatcher.BeginInvoke((Action)(() => drawPoly_XY(new Point(x + x_Offset, y+y_Offset))), null);
961             Dispatcher.BeginInvoke((Action)(() => DrawLine3D(new Point3D(x + x_Offset, y + y_Offset, 0))), null);
962             TrackText(new Point(x + x_Offset, y + y_Offset));
963             DrawCompass(-1*Euler[0], x + x_Offset, y + y_Offset, 100);
964         }
```

ภายใน Function Svt ประกอบด้วยส่วนที่ทำการวาด 2Dและ 3D คือ drawPloy_XY, DrawLine3D, DrawCompass

บทที่ 4
ผลการทดลอง

ในหัวข้อนี้จะกล่าวถึงรายละเอียดผลการทำงานของชิ้นงาน และผลการทำงานของโปรแกรม โดยจะแบ่งการผลทดลองเป็น 3 ส่วนคือ

1. การทดลองการเคลื่อนที่และเคลื่อนไหวด้วยการเดิน
2. การทดลองการเคลื่อนที่ด้วยการติดรถบังคับ
3. การทดลองจำลองการเคลื่อนไหวของวัตถุแบบสามมิติ

4.1 การทดลองวัดการเคลื่อนที่และเคลื่อนไหวด้วยการเดิน

การทดลองนี้ เป็นการทดลองโดยการนำอุปกรณ์ติดกับรองเท้า และทดสอบด้วยการเดิน เพื่อจำลองพฤติกรรมการเดินของมนุษย์การเดิน ทั้งระยะทางในแต่ละก้าว ความสูงของในแต่ละก้าว แบ่งออกเป็นการเดินบนพื้นราบ และเดินขึ้นบันได แสดงผลออกมาเป็นรูปแบบสามมิติ

4.1.1 ทดลองเดินบนพื้นราบ

การทดลองเดินบนพื้นราบ มีจุดประสงค์ เพื่อจำลองพฤติกรรมการเดินบนพื้นราบของมนุษย์และเพื่อทดลองความแม่นยำในการวัดระยะทางของอุปกรณ์ วิธีในการทดลองคือ ติดอุปกรณ์กับรองเท้าเพื่อใช้ทดสอบ ดังรูปที่ 4.1 จากนั้นทดลองเดิน เป็นระยะทาง 810 เซนติเมตร ดังรูปที่ 4.2 จำลองพฤติกรรมการเดิน และวัดระยะทางโดยโปรแกรม ดังรูปที่ 4.3 โดยทำการทดลองซ้ำอีก 10 ครั้ง ได้ผลการทดลอง ตามตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองที่ 4.1.1

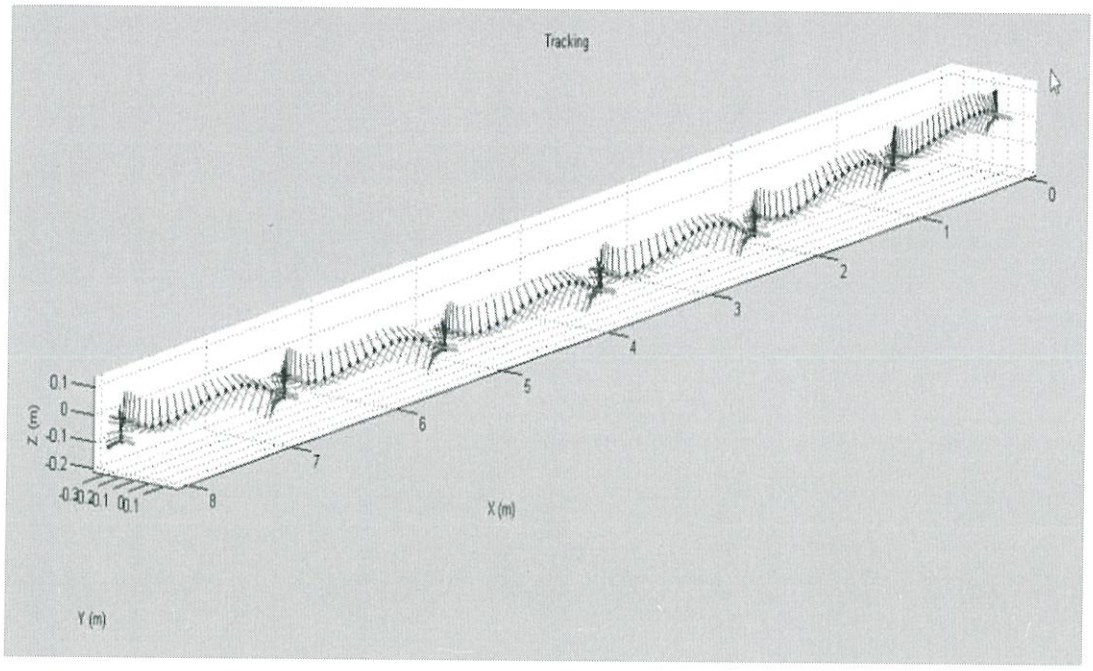
การทดลองครั้งที่	ผลการทดลอง (เซนติเมตร)
1	812
2	807
3	811
4	810
5	812
6	812
7	809
8	814
9	813
10	808

จากการทดลองสรุปได้ว่า อุปกรณ์และโปรแกรม สามารถวัดและจำลองพฤติกรรมการเดินบนพื้นราบของมนุษย์ได้อย่างแม่นยำ ดังแสดงในรูปที่ 4.3 แต่การวัดระยะทาง ยังมีความ

คลาดเคลื่อน แต่อยู่ในระดับที่น้อยมาก ประมาณ ±5 เซนติเมตร เนื่องมาจากประสิทธิภาพของอุปกรณ์

รูปที่ 4.1 ติดอุปกรณ์กับรองเท้าและทดลองเดิน

รูปที่ 4.2 สถานที่ทำการทดลองเดินบนพื้นราบ ระยะทาง 810 เซนติเมตร



รูปที่ 4.3 ภาพจำลองสามมิติ แสดงพฤติกรรมและระยะทางการเดินบนพื้นราบ

4.1.2 ทดลองเดินขึ้นบันได

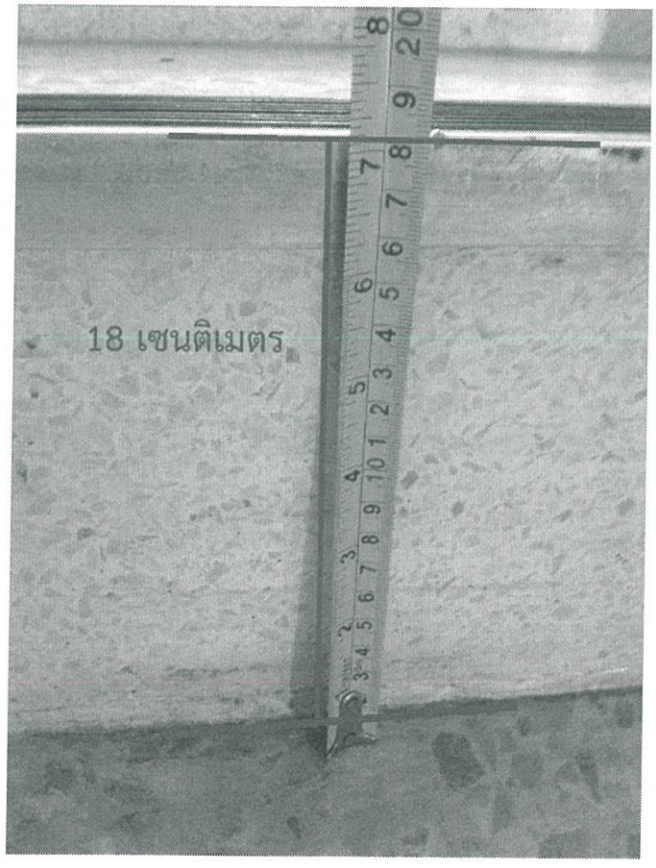
การทดลองเดินขึ้นบันไดนี้ มีจุดประสงค์เพื่อจำลองพฤติกรรมการเดินขึ้นบันไดของมนุษย์ และเพื่อทดลองความแม่นยำในการวัดระยะทางของอุปกรณ์ โดยทำการติดอุปกรณ์กับรองเท้าเพื่อใช้ทดสอบ ดังรูปที่ 4.4 แล้วทดลองเดินขึ้นบันได 24 ขั้น มีความสูงขั้นละ 18 เซนติเมตร ดังรูปที่ 4.5 และทดลองเดินต่อเนื่องบนพื้นราบ ระยะทาง 580 เซนติเมตร ดังรูปที่ 4.6 ทำการจำลองพฤติกรรมการเดิน และวัดระยะ ดังรูปที่ 4.7 ซึ่งทำการทดลองทั้งหมด 10 ครั้ง ได้ผลการทดลองดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองที่ 4.1.2

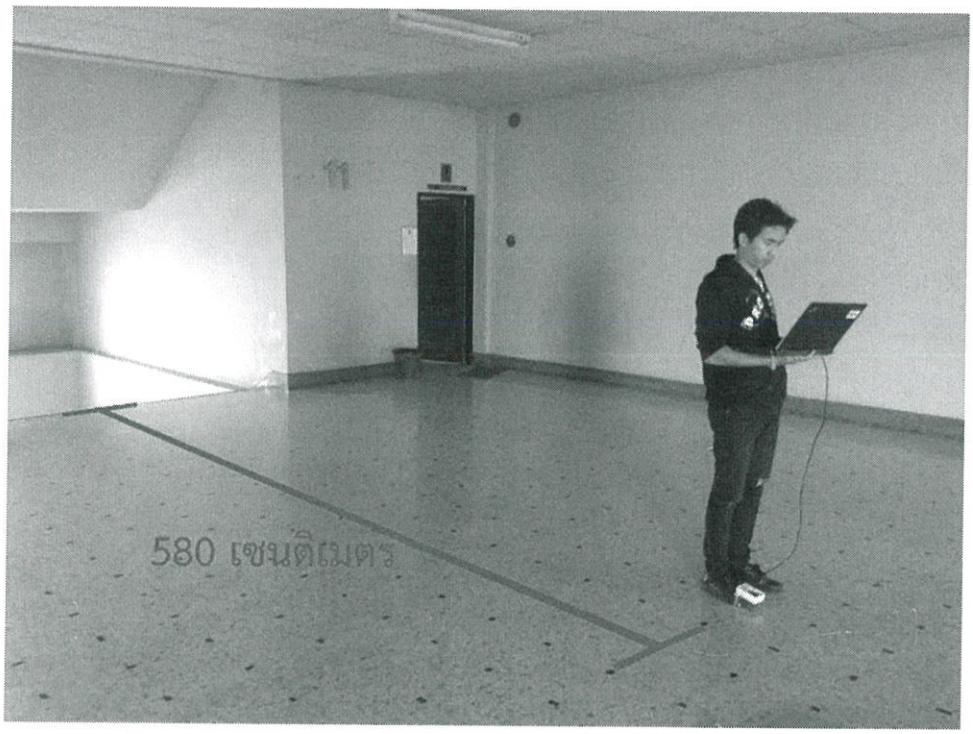
การทดลองครั้งที่	เดินขึ้นบันได (เซนติเมตร)	เดินบนพื้นราบ (เซนติเมตร)
1	434	582
2	428	585
3	432	578
4	433	574
5	434	581
6	434	581
7	427	582
8	424	583
9	438	580
10	430	577

จากการทดลองสรุปได้ว่า อุปกรณ์และโปรแกรม สามารถวัดและจำลองพฤติกรรมการเดินขึ้นบันไดของมนุษย์ได้อย่างแม่นยำ ดังแสดงในรูปที่ 4.7 การวัดระยะทาง ยังมีความคลาดเคลื่อนแต่อยู่ในระดับที่น้อยมาก ประมาณ ±5 เซนติเมตร เนื่องมาจากประสิทธิภาพของอุปกรณ์

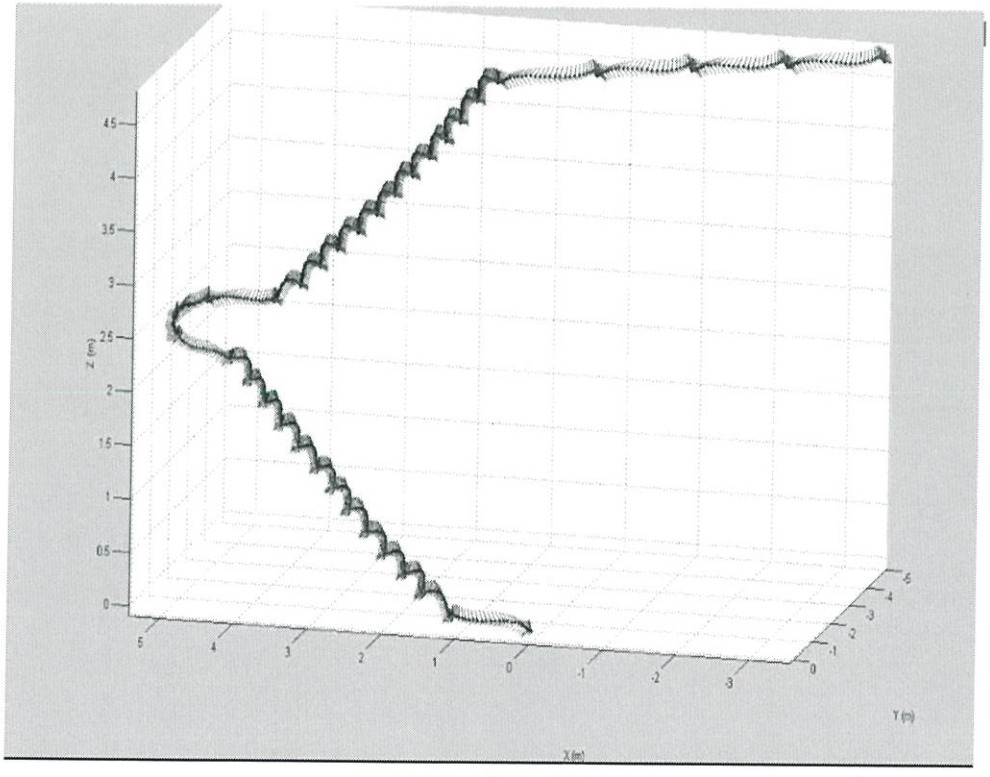
รูปที่ 4.4 ทำการทดลองเดินขึ้นบันได



รูปที่ 4.5 ความสูงของขั้นบันได 18 เซนติเมตร



รูปที่ 4.6 ระยะทางบนเดินต่อบนพื้นราบ 580 เซนติเมตร



รูปที่ 4.7 ภาพจำลองสามมิติ แสดงพฤติกรรมและสเกลการเดินขึ้นบันได

4.2 การทดลองวัดการเคลื่อนที่ด้วยการติดรถบังคับ

การทดลองนี้ เป็นการทดลองโดยนำอุปกรณ์ติดกับรถบังคับ เพื่อวัดการเคลื่อนที่ของวัตถุ ที่มีความเร็วในระดับหนึ่ง อีกทั้งยังสามารถวัดระยะทาง ความสูง ดูพฤติกรรมการเคลื่อนที่ของวัตถุได้ และจำลองเป็นกราฟเส้น แสดงระยะทาง ความสูง พฤติกรรมการเคลื่อนที่ของวัตถุได้

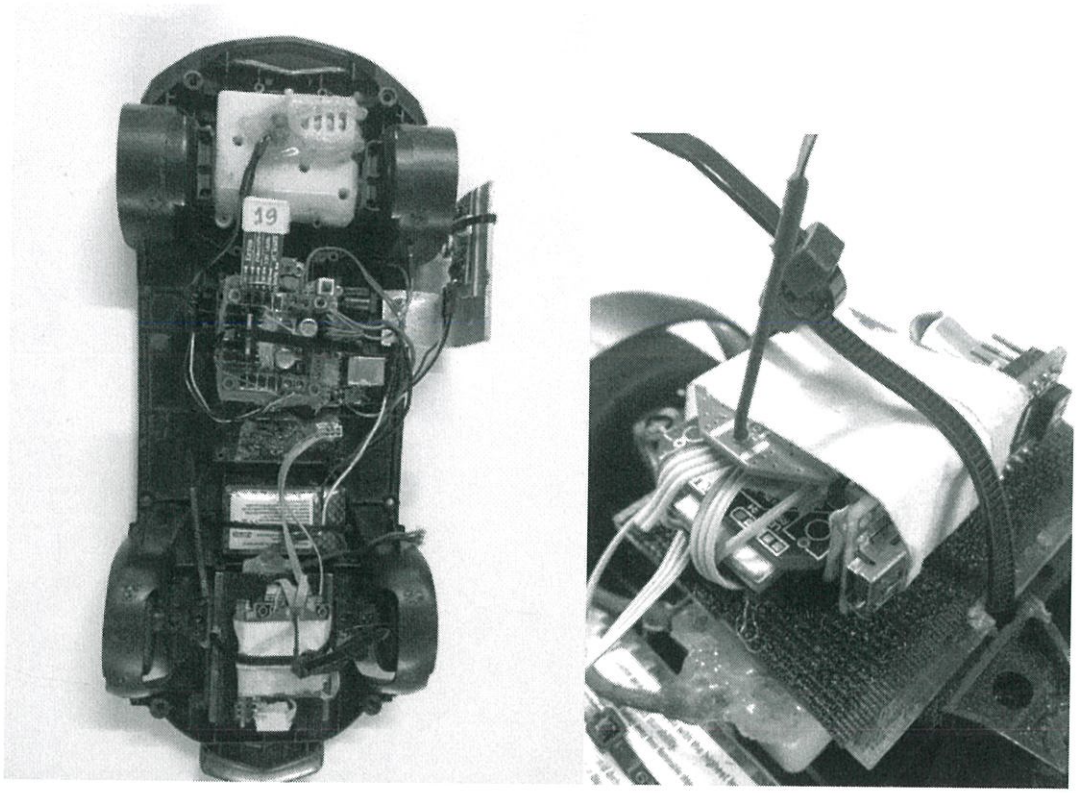
4.2.1 ทดลองวัดระยะทางและความสูง ด้วยการติดอุปกรณ์กับรถบังคับ

การทดลองนี้มีจุดประสงค์เพื่อศึกษาพฤติกรรมการเคลื่อนที่ของวัตถุและเพื่อทดลองความแม่นยำในการวัดระยะทางของอุปกรณ์ โดยวิธีการทดลองคือ ติดอุปกรณ์กับรถบังคับวิทยุ ดังรูปที่ 4.8 ทดลองวิ่งบนสนามทดสอบ ระยะทาง 240 เซนติเมตร ความสูง 15 เซนติเมตร ดังรูปที่ 4.9 และ 4.10 จากนั้นจำลองพฤติกรรมการเคลื่อนที่ของวัตถุ และวัดระยะ ดังรูปที่ 4.11 ซึ่งทำการทดลองทั้งหมด 10 ครั้ง ได้ผลการทดลองดังตางรางที่ 4.3

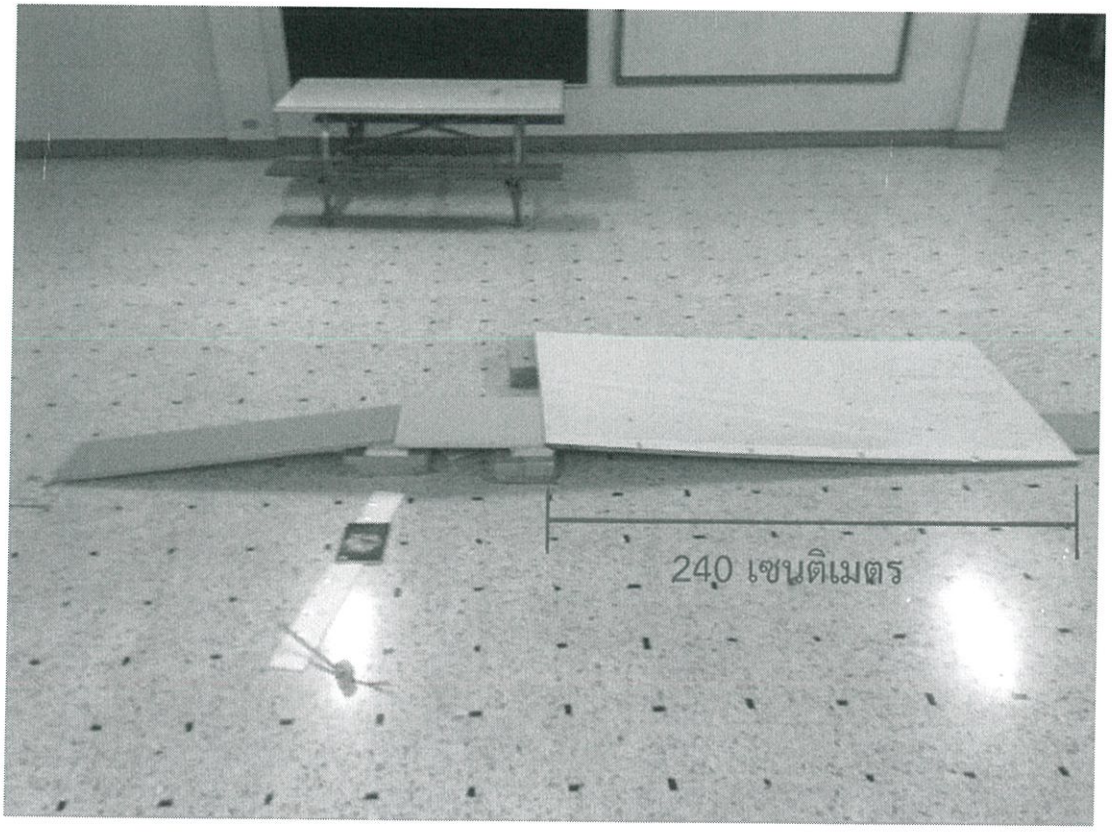
ตารางที่ 4.3 ผลการทดลองที่ 4.2.1

การทดลองครั้งที่	ระยะทาง (เซนติเมตร)	ความสูง (เซนติเมตร)
1	249	14.2
2	247	13.9
3	236	13.4
4	244	13.8
5	245	14.2
6	244	15.6
7	243	15.9
8	244	15.2
9	238	14.8
10	240	14.8

จากการทดลองสรุปได้ว่า อุปกรณ์สามารถแสดงพฤติกรรมการเคลื่อนที่ของวัตถุได้ และการวัดระยะทาง และความสูงยังมีความคลาดเคลื่อน เนื่องจาก ประสิทธิภาพของอุปกรณ์ และการส่งข้อมูลผ่าน Wireless Lan หรือ WiFi ทำให้ข้อมูลผิดเพี้ยน

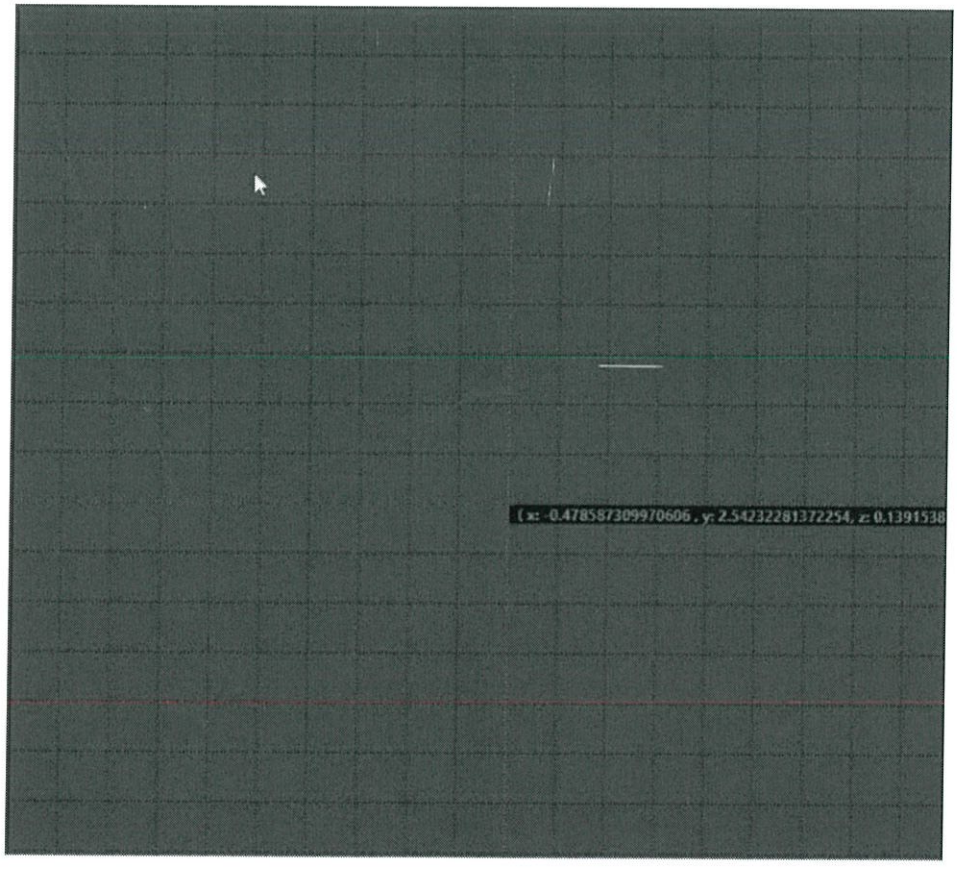


รูปที่ 4.8 รถบังคับที่ติดอุปกรณ์



รูปที่ 4.9 สนามทดลอง และระยะทางทดลองบนแผ่นสีขาว 240 เซนติเมตร

รูปที่ 4.10 ความสูงของสนามทดลอง 15 เซนติเมตร

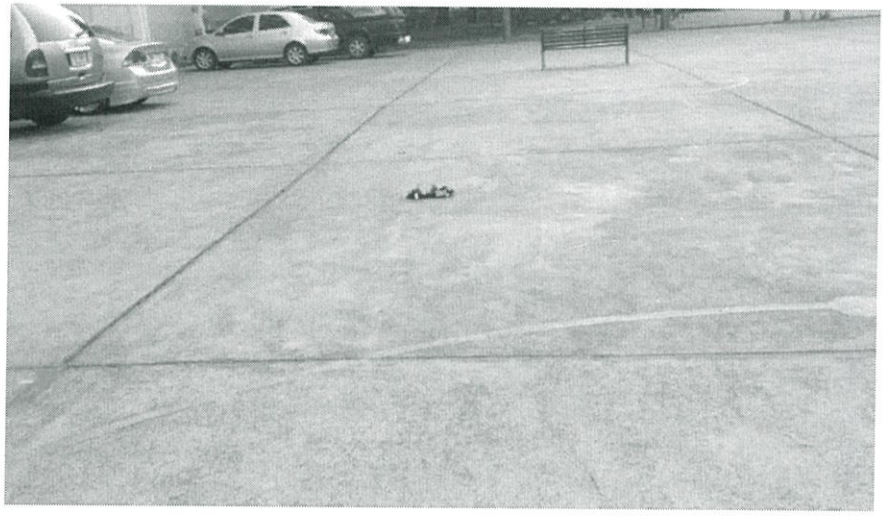


รูปที่ 4.11 กราฟของโปรแกรม แสดงระยะทาง และความสูง ของรถบังคับขณะทำการทดลอง

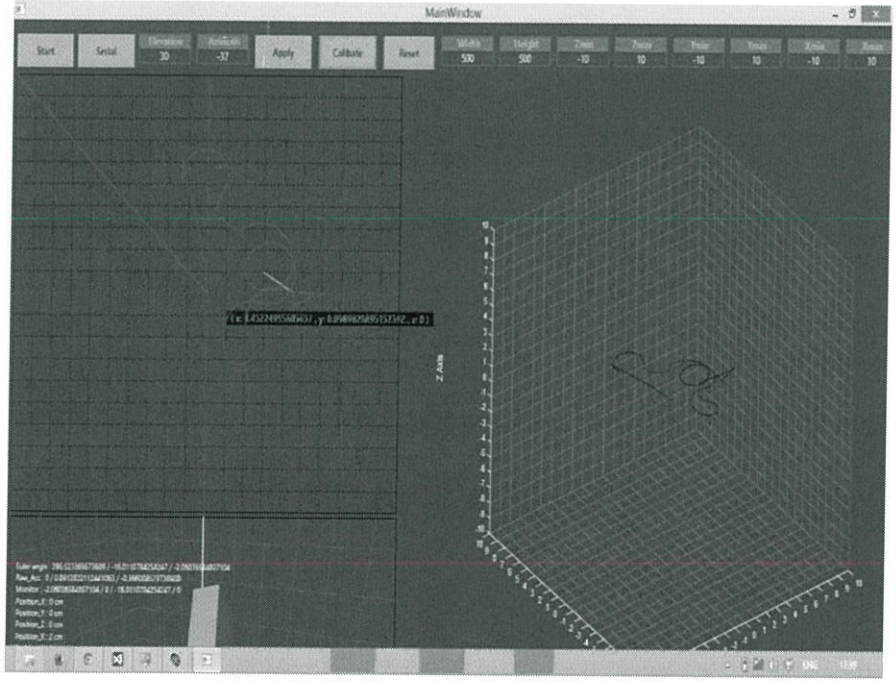
4.2.2 การทดลองบังคับรถบังคับวิทยุ ให้วิ่งอย่างอิสระ

การทดลองนี้มีจุดประสงค์เพื่อจำลองพฤติกรรมการเคลื่อนที่ของวัตถุและเพื่อ ทดลองการส่งข้อมูลผ่าน WiFi โดยวิธีการทดลองคือ ติดอุปกรณ์กับรถบังคับวิทยุ ดังรูปที่ 4.8 บังคับให้รถวิ่งอย่างอิสระ เพื่อจำลองพฤติกรรมการเคลื่อนที่ ดังรูปที่ 4.12 และจำลองพฤติกรรมการเคลื่อนที่ของวัตถุ ดังรูปที่ 4.13

จากการทดลองสามารถสรุปได้ว่า อุปกรณ์และโปรแกรม สามารถจำลองพฤติกรรมการเคลื่อนที่ของรถบังคับได้ สามารถแสดงเป็นเส้นกราฟได้ ว่ารถบังคับมีการเคลื่อนที่อย่างไร



รูปที่ 4.12 สถานที่ทำการทดลองบังคับรถบังคับแบบอิสระ



รูปที่ 4.13 กราฟของโปรแกรม แสดงผลการทดลองบังคับรถบังคับแบบอิสระ

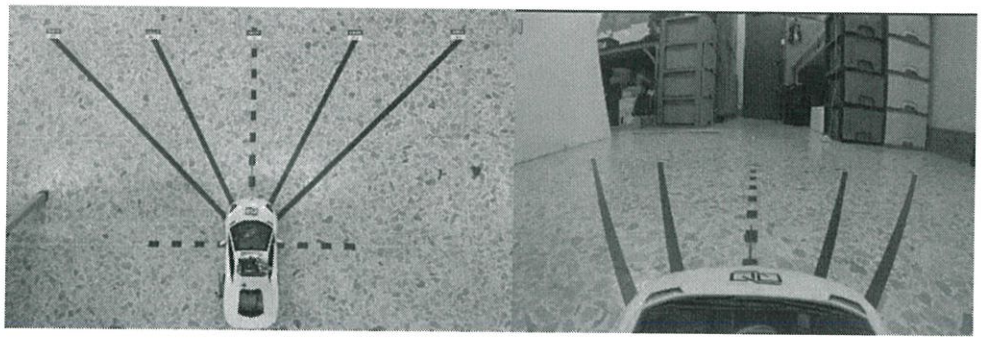
4.2.3 การทดลองวัดมุม

การทดลองนี้มีจุดประสงค์เพื่อวัดมุมที่เปลี่ยนไปเมื่อรถเลี้ยว วิธีการทดลองคือ ตั้งรถให้ตรงเส้นแกนหลัก ดังรูปที่ 4.14 โดยทำการหมุนรถโดยทำมุมจากแกนไปทางซ้ายมือ (มีค่าลบ) -45 และ -22.5 องศา ทำมุมจากแกนไปทางขวามือ (มีค่าบวก) 22.5 และ 45 องศา ดังรูปที่ 4.15,4.16,4.17 และ 4.18 ค่าของมุมจะแสดงผลผ่านโปรแกรม ทำการทดลอง 16 ครั้ง ได้ผลการทดลองดังตารางที่ 4.4 และตัวอย่างผลการทดลองดังรูปที่ 4.19,4.20,4.21 และ 4.22

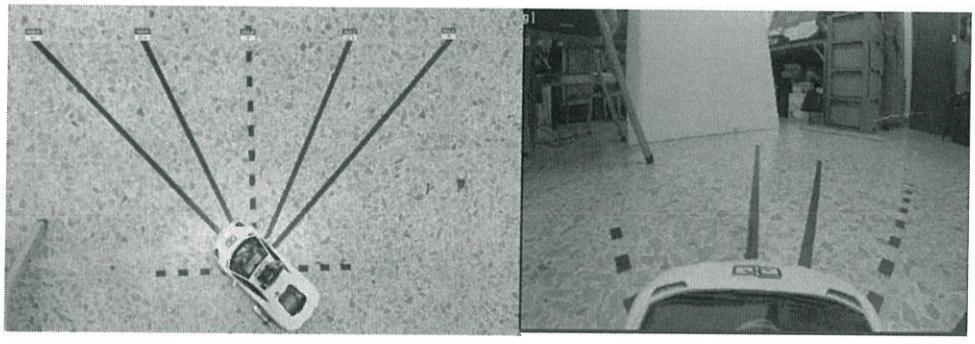
ตารางที่ 4.4 ผลการทดลอง 4.2.3

ครั้งที่	ขนาดมุม	ผลการทดลอง
1	-45	-43.2
2	-22.5	-23.4
3	22.5	22.8
4	45	44.7
5	45	39.3
6	22.5	22.3
7	-22.5	-22.3
8	-45	-44.1
9	-45	-42.4
10	-22.5	-22.2
11	22.5	24.2
12	45	44.8
13	45	45.6
14	22.5	21.6
15	-22.5	-22.3
16	-45	-47.7

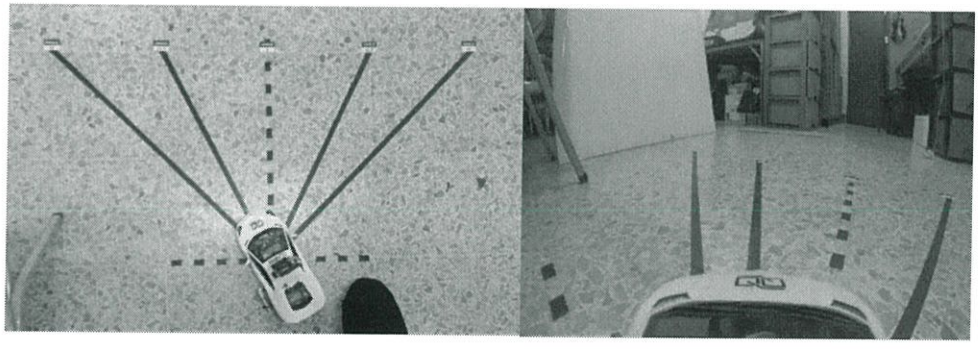
จากผลการทดลองสรุปได้ว่า อุปกรณ์สามารถวัดค่าการเปลี่ยนแปลงของมุมได้ ถึงแม้ค่าจะไม่เที่ยงตรงแม่นยำ แต่มีความคลาดเคลื่อนอยู่ในระดับน้อยมาก โดยเหตุผลที่ทำให้ค่าคลาดเคลื่อน คือ อาจจะหมุนรถทำมุมแตกต่างกันเล็กน้อย และประสิทธิภาพของอุปกรณ์



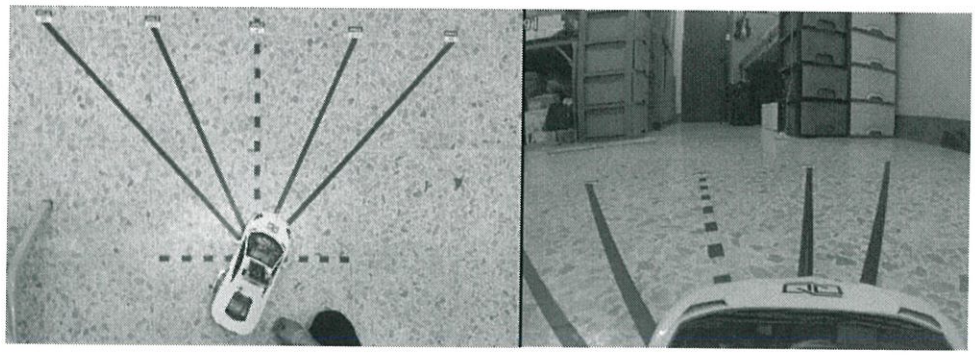
รูปที่ 4.14 ตั้งรถที่แกน 0 องศา



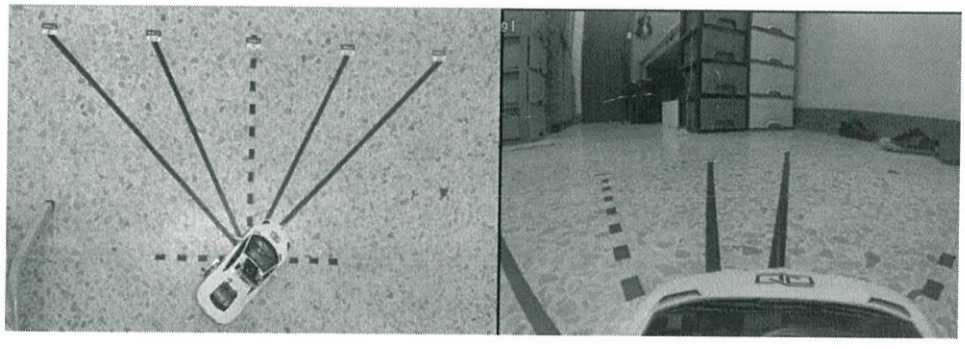
รูปที่ 4.15 หมุนรถทำมุม -45 องศา กับแกน 0 องศา



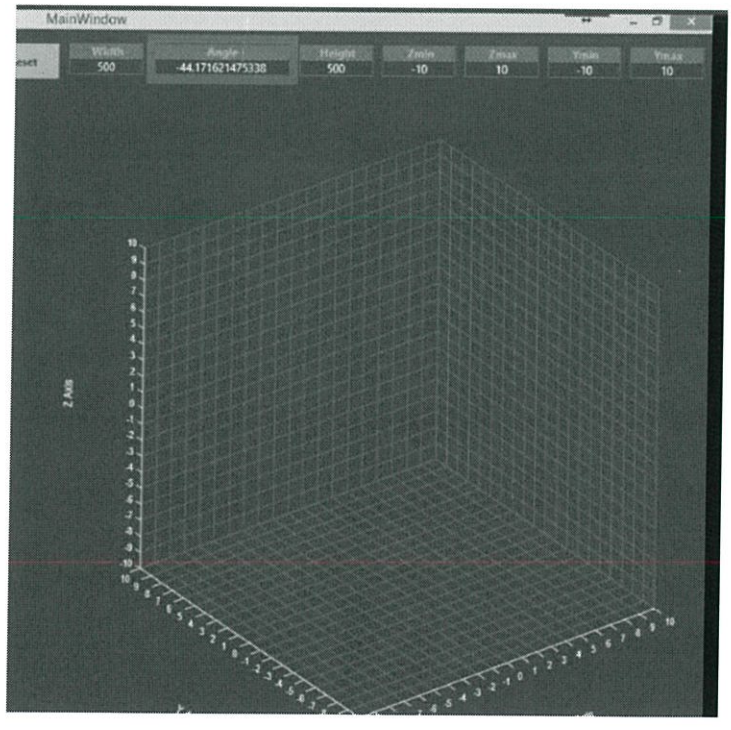
รูปที่ 4.16 หมุนรถทำมุม -22.5 องศา กับแกน 0 องศา



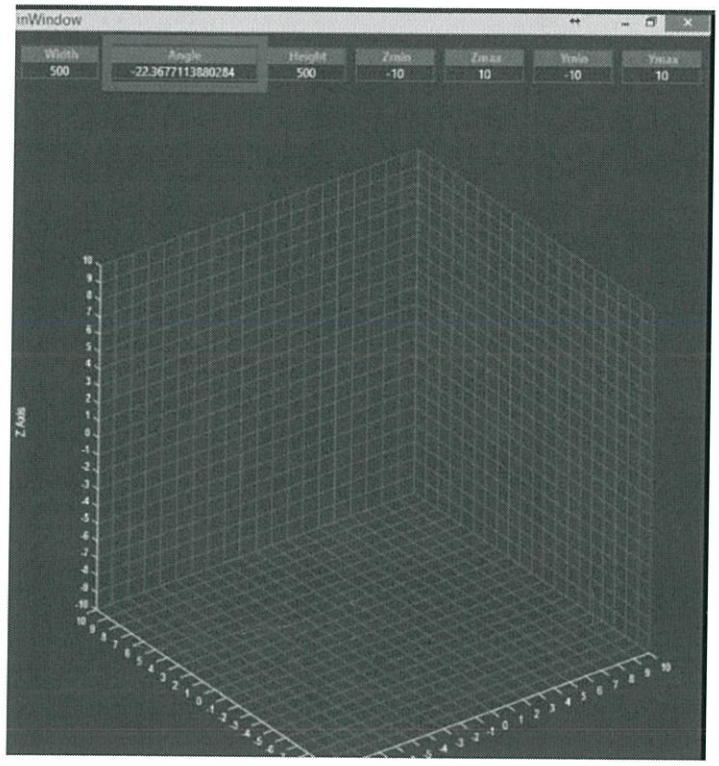
รูปที่ 4.17 หมุนรถทำมุม 22.5 องศา กับแกน 0 องศา



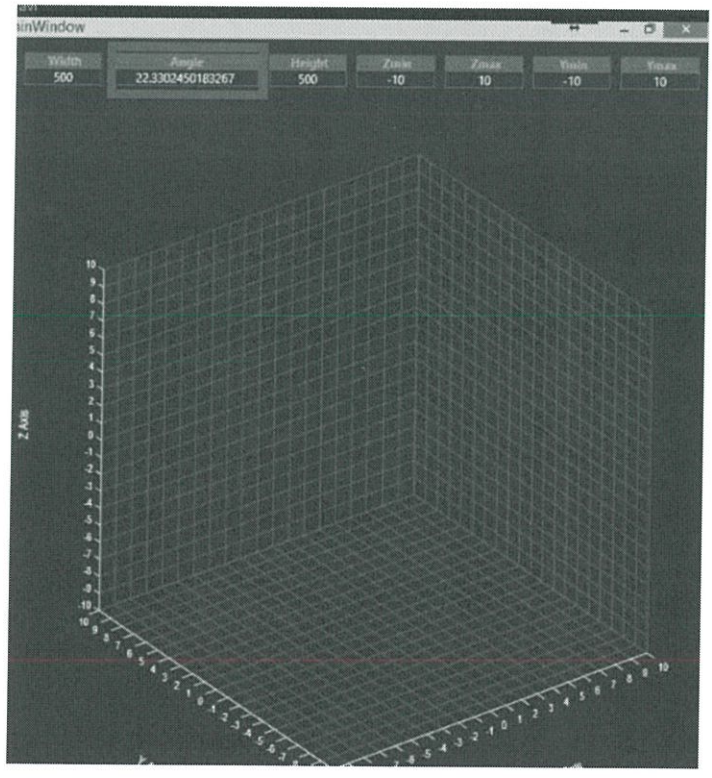
รูปที่ 4.18 หมุนรถทำมุม 45 องศา กับแกน 0 องศา



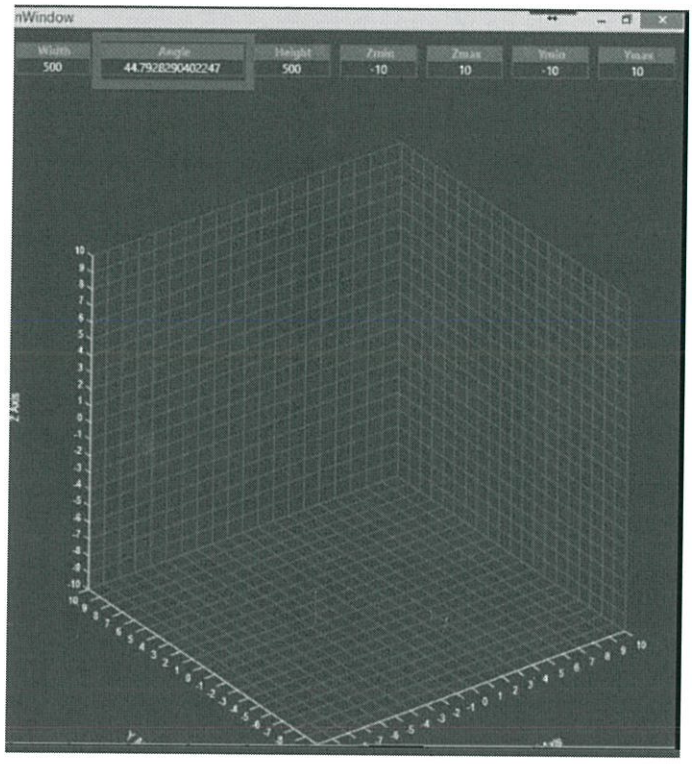
รูปที่ 4.19 ผลการทดลองที่มุม -45 องศา กับแกน 0 องศา



รูปที่ 4.20 ผลการทดลองที่มุม -22.5 องศา กับแกน 0 องศา



รูปที่ 4.21 ผลการทดลองที่มุม 22.5 องศา กับแกน 0 องศา

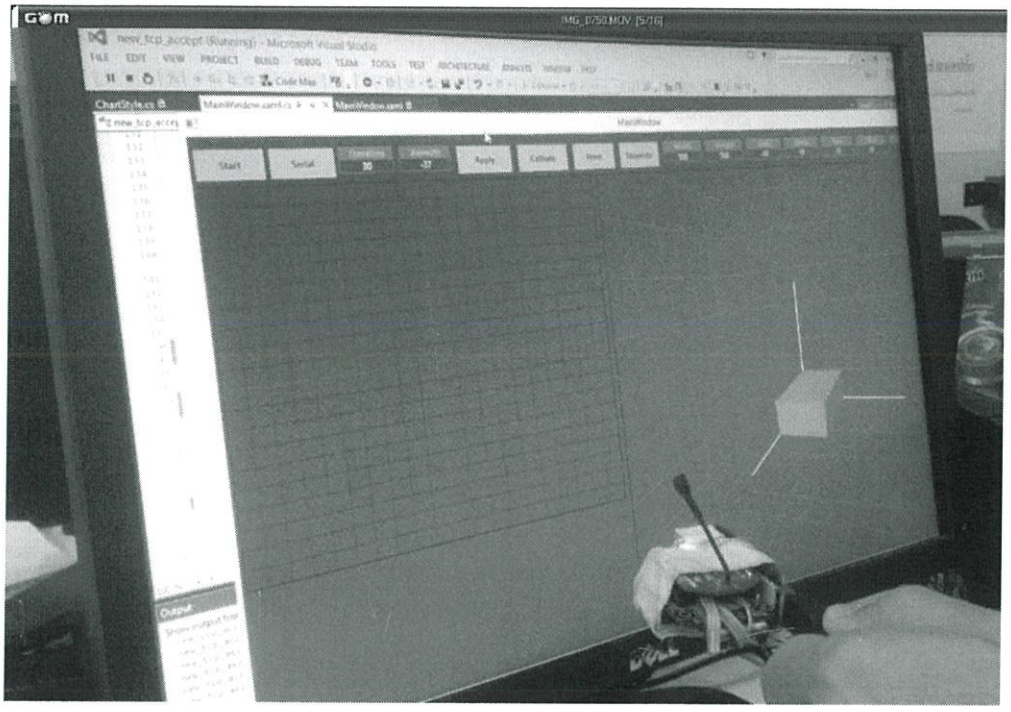


รูปที่ 4.22 ผลการทดลองที่มุม 45 องศา กับแกน 0 องศา

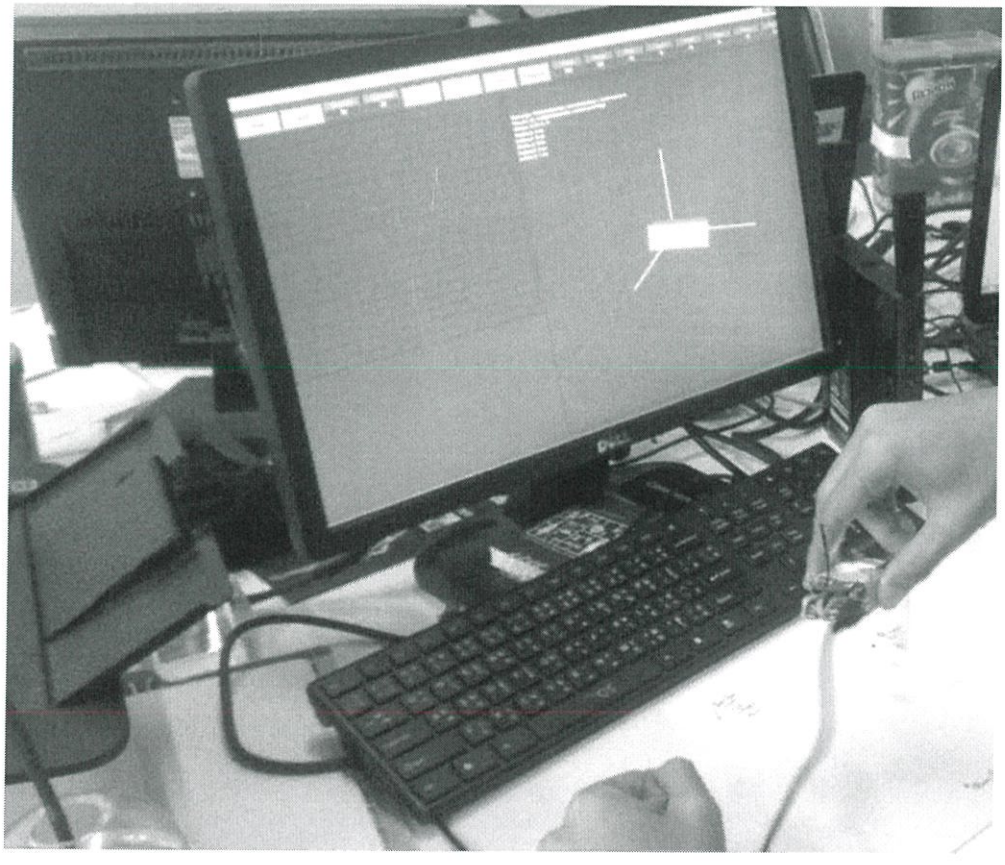
4.3 ผลการทดลองจำลองการเคลื่อนไหวของวัตถุแบบสามมิติ

การทดลองนี้ เป็นการทดลองเพื่อวัดพฤติกรรมการเคลื่อนไหวของวัตถุ และทดลองความแม่นยำในการวัดและส่งค่าข้อมูล โดยการใช้โปรแกรมจำลองการเคลื่อนไหวแบบสามมิติที่พัฒนาขึ้นเอง ร่วมกับอุปกรณ์เซนเซอร์ ทำการทดลองเคลื่อนไหวอุปกรณ์ในอริยาบถต่างๆ ดังรูปที่ 4.23 , 4.24 และ 4.25 แล้วจำลองพฤติกรรมเป็นรูปแบบสามมิติ

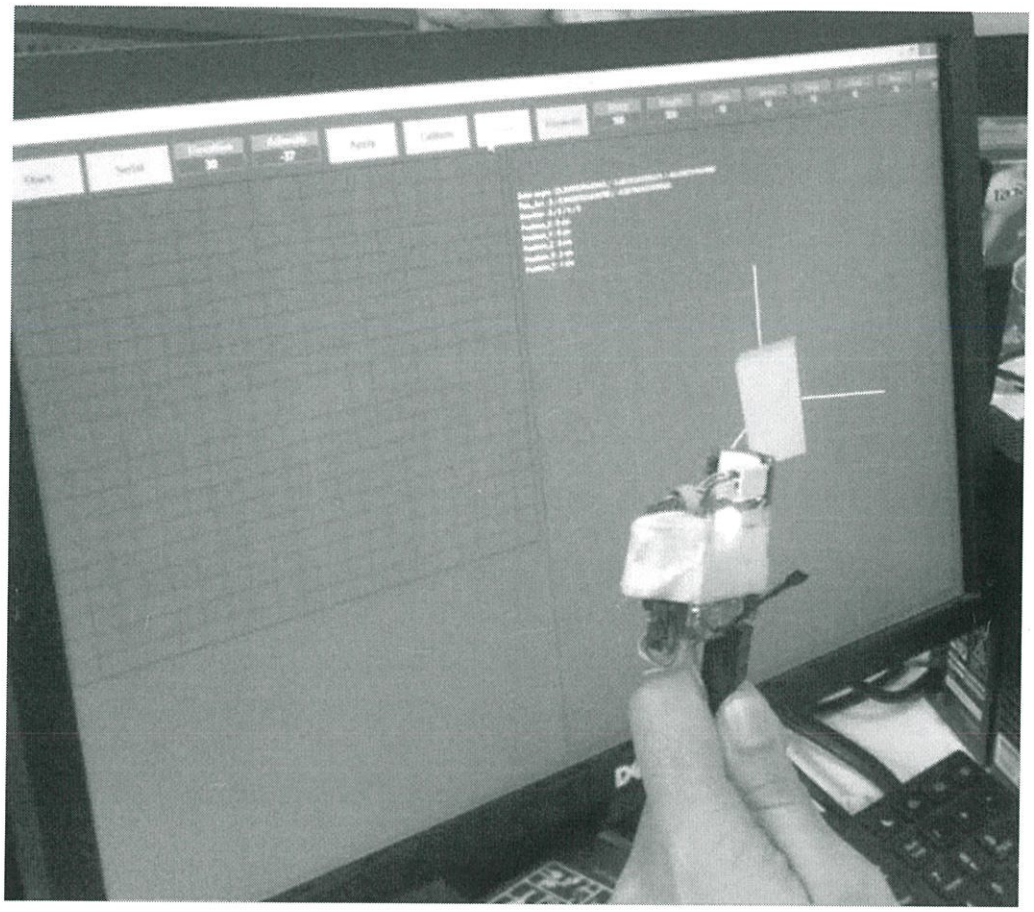
จากการทดลองสรุปได้ว่า อุปกรณ์และโปรแกรมสามารถแสดงพฤติกรรมอริยาบถการเคลื่อนไหวของวัตถุได้ แต่ยังมีความคลาดเคลื่อน ในกรณีที่มีการเปลี่ยนอริยาบถเร็วๆ เนื่องมาจากความละเอียดและประสิทธิภาพของตัวอุปกรณ์



รูปที่ 4.23 อุปกรณ์อยู่ระนาบปกติ ก่อนทำการเคลื่อนไหว



รูปที่ 4.24 ทดลองโดยการหมุนอุปกรณ์ในแนวระนาบ



รูปที่ 4.25 ทดลองโดยการตั้งอุปกรณ์ขึ้น

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

5.1 บทสรุปโครงงาน

โครงงานนี้ นำเสนอในเรื่องของการพัฒนาอุปกรณ์เพื่อใช้ในการวัดการเคลื่อนที่และเคลื่อนไหวของวัตถุในรูปแบบสามมิติ โดยใช้หลักการ วัดค่าความเร่ง วัดความเร็วเชิงมุม และวัดทิศทางของวัตถุ มาเข้ากระบวนการ และจำลองว่าวัตถุมีการเคลื่อนที่อย่างไร มีการเคลื่อนไหวอย่างไร ในการทดลองการทำงานของระบบจากการทดลองในบทที่ 4 นั้น ได้ทำการทดลองการเคลื่อนที่และเคลื่อนไหวแบบต่างๆ ซึ่งมีการใช้อุปกรณ์ทดลองที่สร้างขึ้นและโปรแกรมที่พัฒนาขึ้นเอง จากการทดลองนั้น ผลการทดลองที่ออกมาคือ สามารถวัดการเคลื่อนที่ การเคลื่อนไหวของวัตถุ และศึกษาพฤติกรรมได้ แต่ยังมีความไม่แม่นยำเหลืออยู่บ้างเพียงเล็กน้อย และยังมีการแสดงผลที่ผิดพลาดอยู่บ้าง

5.2 ปัญหาที่พบในระหว่างดำเนินงาน

1. การแสดงผลยังไม่แม่นยำและเที่ยงตรงบ้าง
2. ส่วนประกอบในอุปกรณ์บางชิ้นส่วน ไม่มีประสิทธิภาพเพียงพอในการตอบสนองการเขียนโปรแกรม
3. การนำอุปกรณ์ไปติดกับวัตถุที่ต้องการวัดการเคลื่อนที่ มีข้อจำกัดด้านความเข้ากันได้กับวัตถุนั้นๆ
4. มีข้อจำกัดด้านขีดความสามารถของอุปกรณ์ ที่ให้ค่าไม่ละเอียดเท่าที่ควร อันเนื่องมาจากต้นทุนการผลิต
5. อุปกรณ์มีขนาดใหญ่และมีชิ้นส่วนที่มากเกินความจำเป็น

5.3 แนวทางแก้ไขและพัฒนาต่อ

1. ปรับปรุงการแสดงผลและวัดค่าให้มีความเที่ยงตรงแม่นยำ
2. พัฒนาและลดขนาดของอุปกรณ์ให้มีขนาดเล็กลง และเหลือเพียงชิ้นส่วนที่มีความจำเป็น
3. ต้องมีทุนมากกว่านี้ เพื่อสามารถจัดหาอุปกรณ์ที่มีขีดความสามารถสูงขึ้นได้

บรรณานุกรม

[1] Wireless Lan (2556) เข้าถึงได้จาก : http://portal.in.th/eleccom57/pages/6042/ (วันที่ค้นข้อมูล : 28 สิงหาคม 2556).

[2] Wireless Lan (2550) เข้าถึงได้จาก : http://blog.eduzones.com/banny/3481 (วันที่ค้นข้อมูล : 2 กันยายน 2556).

[3] Wireless Lan (ม.ป.ป.) เข้าถึงได้จาก : http://www.cc.kmutt.ac.th/wireless/ about.html (วันที่ค้นข้อมูล : 2 กันยายน 2556).

[4] Wireless Lan (ม.ป.ป.) เข้าถึงได้จาก : http://www.phrae.mju.ac.th/ITS/page4.asp (วันที่ค้นข้อมูล : 3 กันยายน 2556).

[5] Wireless Lan (ม.ป.ป.) เข้าถึงได้จาก : http://center.finansalife.com/qagt/ wirelesslan.htm (วันที่ค้นข้อมูล : 4 กันยายน 2556).

[6] ทฤษฎีการเคลื่อนที่ (ม.ป.ป.) เข้าถึงได้จาก : http://www.rmutphysics.com/charud/ scibook/ dynamic1/index6_1.html (วันที่ค้นข้อมูล : 4 กันยายน 2556).

[7] บอร์ดไมโครคอนโทรลเลอร์ (ม.ป.ป.) เข้าถึงได้จาก : http://arduino.cc/en/Main/ ArduinoBoardNano (วันที่ค้นข้อมูล : 2 มิถุนายน 2556).

[8] ไมโครคอนโทรลเลอร์ (ม.ป.ป.) เข้าถึงได้จาก : http://www.alldatasheet.com /datasheet- pdf/pdf/392243/ATMEL/ATMEGA328.html (วันที่ค้นข้อมูล : 2 มิถุนายน 2556).

[9] Accelerometer (ม.ป.ป.) เข้าถึงได้จาก : http://en.wikipedia.org/wiki/ Accelerometer(วันที่ค้นข้อมูล : 2 มิถุนายน 2556).

[10] Accelerometer& Gyroscope (ม.ป.ป.) เข้าถึงได้จาก http://www.spicydog.org/ board/index.php?topic=431.0 (วันที่ค้นข้อมูล : 2 มิถุนายน 2556).

[11] Gyroscope (ม.ป.ป.) เข้าถึงได้จาก : http://en.wikipedia.org/wiki/Gyroscope (วันที่ค้นข้อมูล : 2 มิถุนายน 2556).

[12] ทฤษฎีการหมุนของวัตถุ (ม.ป.ป.) เข้าถึงได้จาก http://www.space.mict.go.th/knowledge.php?id=navapp3 (วันที่ค้นข้อมูล : 2 สิงหาคม 2556).

[13] โปรแกรม Processing (ม.ป.ป.) เข้าถึงได้จาก : https://sites.google.com/site/ eplearn/processing (วันที่ค้นข้อมูล : 27 สิงหาคม 2556).

[14] 3D Coordinate (ม.ป.ป.) เข้าถึงได้จาก : http://msdn.microsoft.com/en-us/library/ms747437.aspx (วันที่ค้นข้อมูล : 15 สิงหาคม 2556).

[15] เข็มทิศ เข้าถึงได้จาก : http://th.wikipedia.org/wiki/เข็มทิศ (วันที่ค้นข้อมูล : 15 สิงหาคม 2556).

[16] Logic Level เข้าถึงได้จาก : http://en.wikipedia.org/wiki/Logic_level (วันที่ค้นข้อมูล : 15 สิงหาคม 2556).

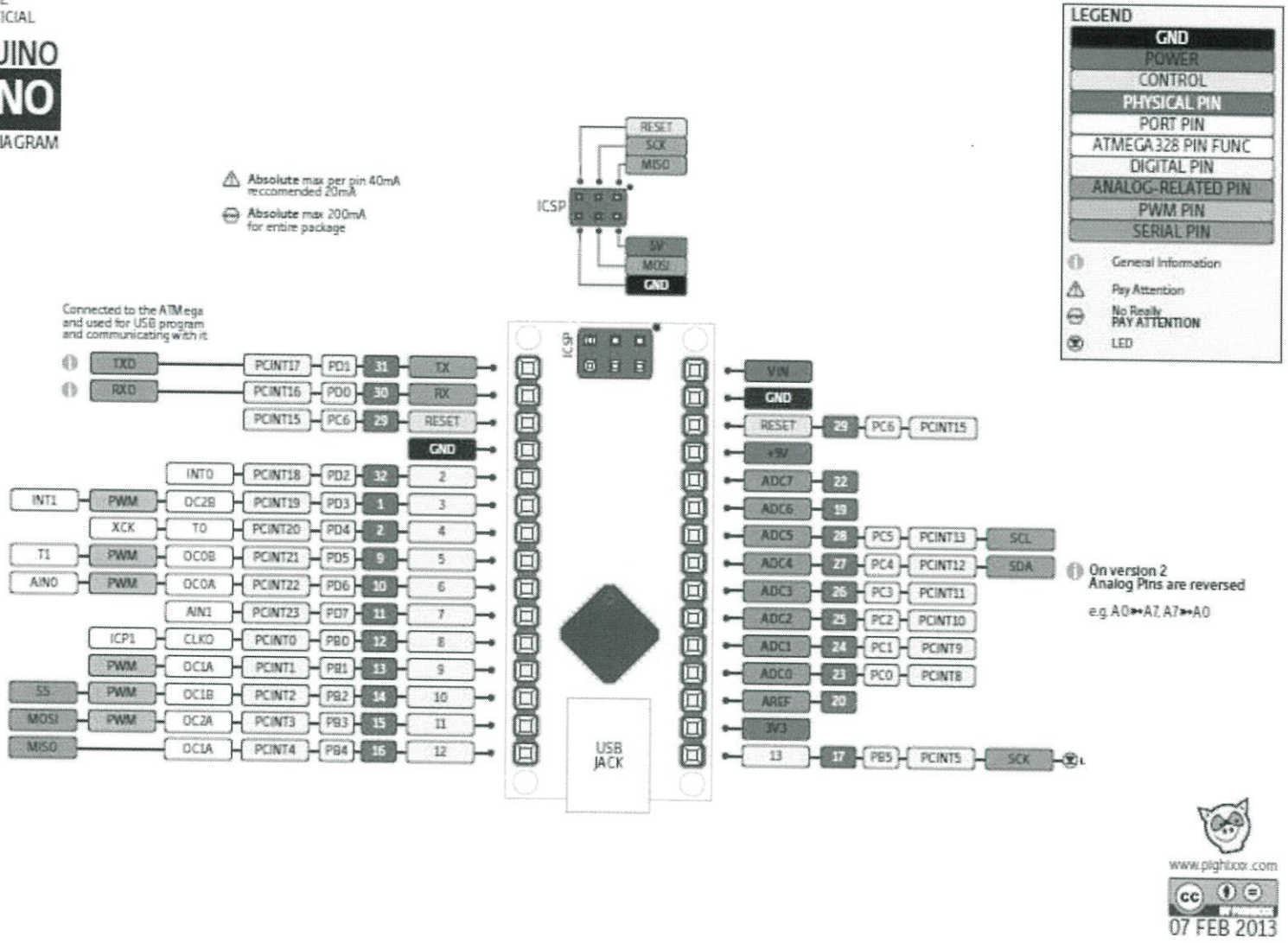
[17] IR Sensor เข้าถึงได้จาก : http://nakasut007ster1234.blogspot.com /2006/12/infared-light.html (วันที่ค้นข้อมูล : 15 สิงหาคม 2556).

ภาคผนวก

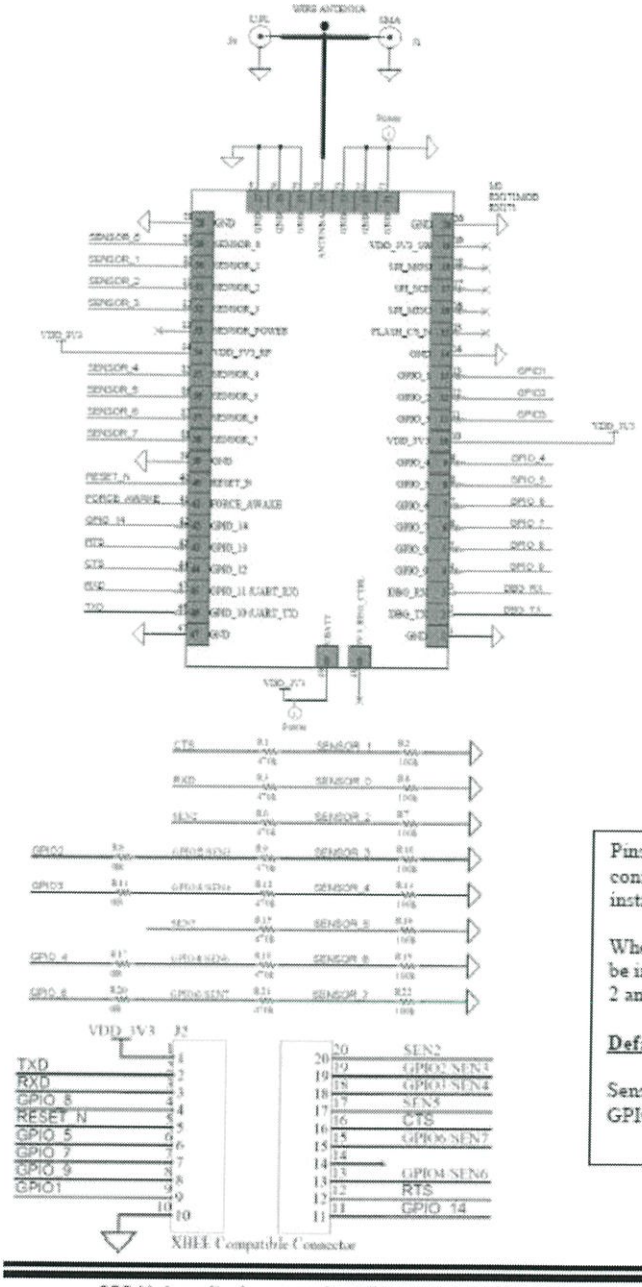
THE
UNOFFICIAL

ARDUINO NANO

PINOUT DIAGRAM



Module Schematic



Pins 13, 15, 18 and 19 on the RN-XV connector can be configured as GPIOs or as sensor inputs depending on the installed resistors.

When configured as GPIOs, ONLY Column 1 resistors MUST be installed. When configured as sensor inputs, ONLY Column 2 and Column 3 resistors MUST be stuffed.

Default configuration:

Sensor inputs – Pins 19 and 20
GPIO – Pins 13, 15 and 18

UART Interface
TTL signals

RX - input to SuRF board
TX - output from SuRF board

(Note: the labels on the board for RX and TX are incorrect)

PIN	Description
1	3.3 VDC
2	GND
3	RXDB
4	TXDB
5	RTSB
6	CTSB
7	PIO4
8	PIO5
9	PIO6
10	PIO7
11	PIO8
12	Reset

Evaluation Board Connector

Power

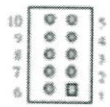
5-12 VDC
GND

Sensors

WARNING: Sensors must not be driven with more than 1.2 VDC or permanent damage will occur

PIN	Description
1	Sensor PWR
2	Resistor network to Sensor4
3	Resistor network to Sensor5
4	Sensor7 (1.2 VDC)
5	Sensor4 (1.2 VDC)
6	Sensor5 (1.2 VDC)
7	Sensor6 (1.2 VDC)
8	GND

RS232 Interface



RX - input to SuRF board
TX - output from SuRF board

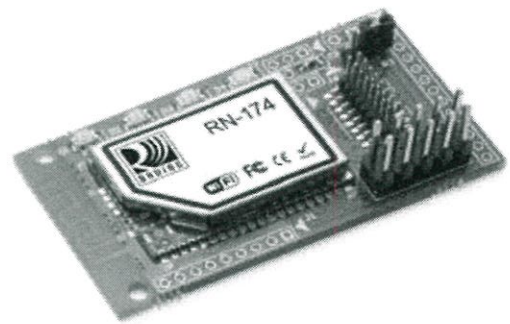
PIN	Description
1	No connect
2	RX
3	TX
4	No connect
5	GND
6	No connect
7	RTS
8	CTS
9	5 – 12 VDC
10	No connect

Jumper Block 1 - 4

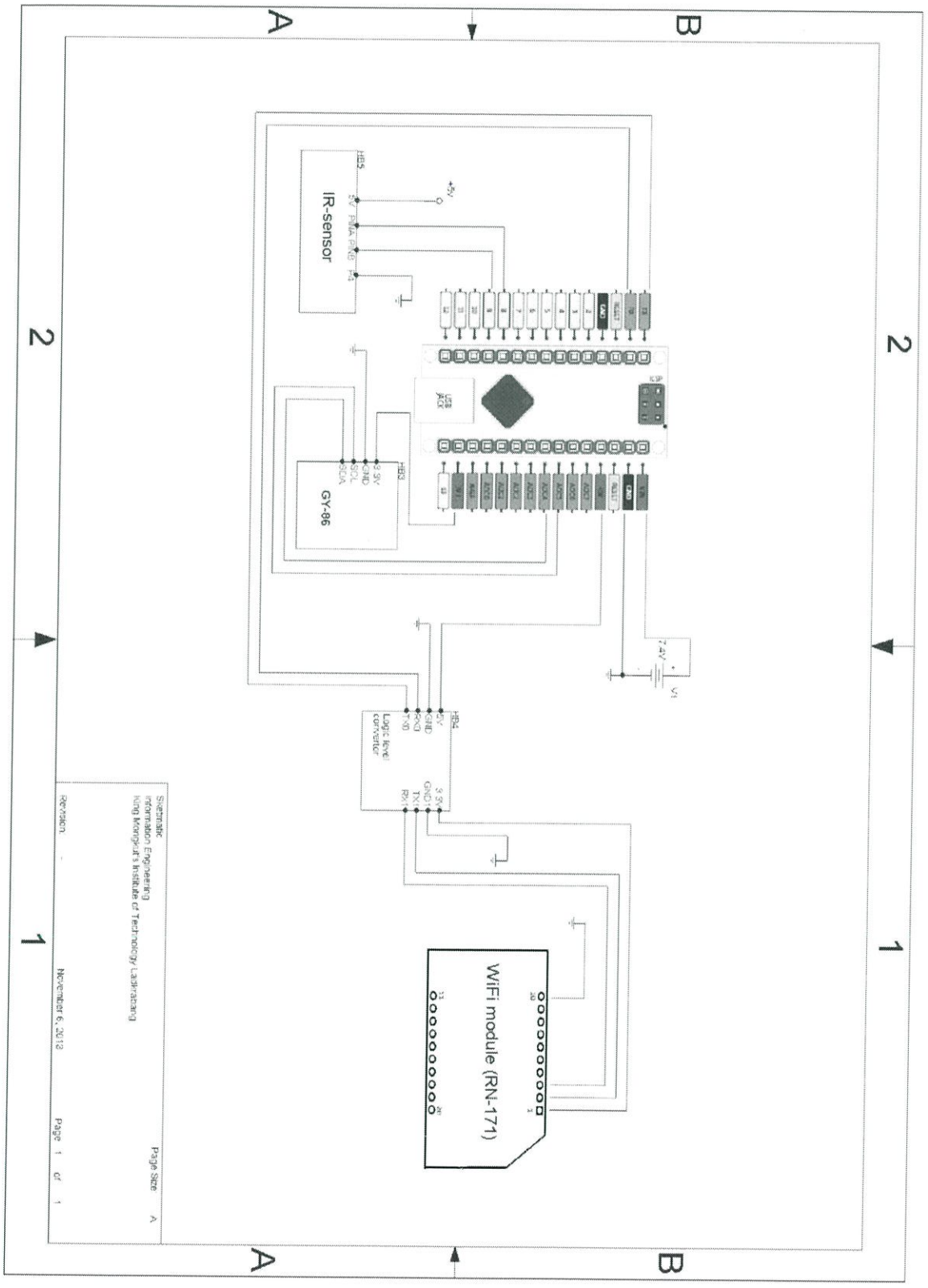
1) Adhoc mode & Factory Reset
2) Config 1
3) Config 2
4) Config 3

LED Indicators

Condition	Blue LED	Red LED	Yellow LED	Green LED
ON solid	Power On			Connected over TCP
Fast blink		Not Associated	RX/TX data transfer	No IP address
Slow blink		Associated, No Internet		IP address OK
OFF	No Power	Associated Internet OK		



ลายวงจรของอุปกรณ์ที่พัฒนาขึ้น



Schematic ของ Motion Sensor